
STPC[®] Vega Programming Manual

Release B

March 6, 2003



STMicroelectronics

Information provided is believed to be accurate and reliable. However, ST Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringements of patents or other rights of third parties which may result from its use. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied.

1 Table of contents

1 Table of contents	3
2 List of Tables	13
3 List of Figures	15
4. HOW TO USE THIS MANUAL	17
4.1. INTRODUCTION	17
4.2. SPECIFIC NOTES	17
4.2.1. RESERVED BITS	17
4.2.2. SIGNAL ACTIVE STATE	17
4.2.3. HEXADECIMAL NOTATION	17
4.2.4. ENDIAN	18
4.3. ISSUING NOTES	19
5 LIST OF REGISTERS	21
6. HOST INTERFACE	33
6.1. INTRODUCTION	33
6.2. AGENT DECODING	35
6.3. MEMORY ADDRESS MAP	35
6.3.1. MEMORY HOLE	37
6.3.2. SMM MEMORY	37
6.3.3. ADDRESSABLE SDRAM MEMORY	37
6.3.4. CPU ADDRESS TO SDRAM ADDRESS MAPPING	37
6.4. IO ADDRESS MAP	39
6.4.1. PCI CONFIGURATION ADDRESS MAP:	39
6.5. CACHE RELATED REGISTERS	40
6.5.1. CACHE ARCHITECTURE REGISTER 0	40
6.5.2. CACHE ARCHITECTURE REGISTER 1	42
6.5.3. CACHE ARCHITECTURE REGISTER 2	43
6.6. ADDRESS DECODE RELATED REGISTERS	45
6.6.1. MEMORY HOLE CONTROL REGISTER	45
6.6.2. SHADOW CONTROL REGISTER 0	46
6.6.3. SHADOW CONTROL REGISTER 1	48
6.6.4. SHADOW CONTROL REGISTER 2	49
6.6.5. SHADOW CONTROL REGISTER 3	50
6.7. HOST SDRAM CONTROLLER REGISTERS	52
6.7.1. MEMORY BANK 0 REGISTER - C.I. 30H (Memory_Bank0)	52
6.7.2. MEORY BANK 1 REGISTER - C.I. 31H (Memory_Bank1)	52



6.7.3. MEMORY BANK 2 REGISTER - C.I. 32H (Memory_Bank2)	52
6.7.4. MEMORY BANK 3 REGISTER - C.I. 33H (Memory_Bank3)	52
6.7.5. SDRAM REFRESH REGISTER	53
6.8. ACCESSING CONFIGURATION REGISTERS	54
6.1 GPCLK CONTROL REGISTERS	55
6.1.1. GPCLK Control Register 00	55
6.1.2. GPCLK control register 01	56
6.1.3. GPCLK control register 10	57
6.1.4. GPCLK control register 11	58
6.1.5. GPCLK control register 20	59
6.1.6. GPCLK control register 21	60
6.1.7. GPCLK control register 30	61
6.1.8. GPCLK control register 31	62
7. SDRAM Controller	63
7.1. INTRODUCTION	63
7.2. MEMORY CONTROLLER	63
7.3. SDRAM REGISTER ACCESS	64
7.3.1. REGISTER 0	64
7.3.2. REGISTER 1	66
7.3.3. REGISTER 2 (MEM_REG2) 84C6008h	67
7.3.4. DDC CONTROL REGISTER DDCR	68
7.4. MEMORY CLOCK REGISTERS	69
7.4.1. MCLK control register 0	69
7.4.2. MCLK control register 1	70
8. PCI CONTROLLERS	71
8.1. INTRODUCTION	71
8.1.1. PCI ADDRESS DECODE	72
8.1.2. PCI ERROR HANDLING	72
8.1.3. PCI ARBITER	72
8.2. ACCESSING THE PCI CONFIGURATION REGISTERS	72
8.3. PCI CONFIGURATION ADDRESS REGISTER	73
8.4. CONFIGURATION DATA REGISTER	74
8.5. NORTH BRIDGE CONFIGURATION REGISTERS	74
8.5.1. NORTH BRIDGE PCI COMMAND REGISTER	75
8.5.2. NORTH BRIDGE PCI STATUS REGISTER	76
8.5.3. NORTH BRIDGE PCI REVISION ID REGISTER	77

8.5.4. NORTH BRIDGE DEVICE CLASS CODE REGISTER	78
8.5.5. NORTH BRIDGE HEADER TYPE REGISTER	79
8.5.6. NORTH BRIDGE CONTROL REGISTER	80
8.5.7. NORTH BRIDGE PCI ERROR STATUS REGISTER	81
8.6. THE SOUTH BRIDGE	82
8.7. SOUTH BRIDGE PCI ISA CONFIGURATION REGISTERS	82
8.7.1. SOUTH BRIDGE PCI COMMAND REGISTER	83
8.7.2. SOUTH BRIDGE PCI STATUS REGISTER	84
8.7.3. SOUTH BRIDGE PCI REVISION ID REGISTER	85
8.7.4. SOUTH BRIDGE DEVICE CLASS CODE REGISTER	86
8.7.5. SOUTH BRIDGE HEADER TYPE REGISTER	87
8.7.6. SOUTH BRIDGE MISCELLANEOUS REGISTER	88
8.8. PCI TO IDE BRIDGE CONFIGURATION REGISTERS	89
8.8.1. PCI to IDE BRIDGE PCI COMMAND REGISTER	90
8.8.2. PCI to IDE BRIDGE PCI STATUS REGISTER	91
8.8.3. PCI to IDE BRIDGE REVISION ID REGISTER	92
8.8.4. PCI to IDE BRIDGE PROGRAMMING INTERFACE REGISTER	93
8.8.5. PCI to IDE BRIDGE SUB-CLASS CODE REGISTER	94
8.8.6. PCI to IDE BRIDGE BASE-CLASS CODE REGISTER	95
8.8.7. PCI to IDE BRIDGE LATENCY TIMER CONTROL REGISTER	96
8.8.8. PCI to IDE BRIDGE HEADER TYPE REGISTER	97
8.8.9. PCI to IDE BRIDGE IDE BASE ADDRESS 0 REGISTER	98
8.8.10. PCI to IDE BRIDGE IDE BASE ADDRESS 1 REGISTER	99
8.8.11. PCI to IDE BRIDGE IDE BASE ADDRESS 2 REGISTER	100
8.8.12. PCI to IDE BRIDGE IDE BASE ADDRESS 3 REGISTER	101
8.8.13. PCI to IDE BRIDGE IDE BASE ADDRESS 4 REGISTER	102
8.8.14. MISC REGISTER 0	103
8.8.15. MISC REGISTER 1	104
8.8.16. PRIMARY MASTER TIMING REGISTER	105
8.8.17. PRIMARY SLAVE TIMING REGISTER	106
8.8.18. SECONDARY MASTER TIMING REGISTER	107
8.8.19. SECONDARY SLAVE TIMING REGISTER	108
8.9. PCI TO USB BRIDGE CONFIGURATION REGISTERS	109
8.9.1. PCI to USB BRIDGE VENDOR IDENTIFICATION REGISTER	110
8.9.2. PCI to USB BRIDGE DEVICE IDENTIFICATION REGISTER	111
8.9.3. USB BRIDGE PCI COMMAND REGISTER	112
8.9.4. USB BRIDGE PCI STATUS REGISTER	113
8.9.5. USB BRIDGE PCI REVISION ID REGISTER	114
8.9.6. USB BRIDGE DEVICE CLASS CODE REGISTER	115
8.9.7. USB BRIDGE HEADER TYPE REGISTER	116

8.10. PCI CONFIGURATION FOR OPENHCI-COMPLIANT USB HOST CONTROLLER	117
8.10.1. COMMAND REGISTER	118
8.10.2. CLASS_CODE Register	119
8.10.3. BAR_OHCI Register	120
8.11. LEGACY USB SUPPORT REGISTERS	122
8.12. PCI TO LAN BRIDGE CONFIGURATION REGISTERS	123
8.12.1. Command Reg	124
8.12.2. Status Register	125
8.12.3. Revision ID	126
8.12.4. Class Code	126
8.12.5. Cache Line Size	127
8.12.6. Latency Timer	127
8.12.7. Header Type	127
8.12.8. BIST	127
8.12.9. Memory Base Address 0	128
8.12.10. I/O Base Address 0	129
8.12.11. Interrupt Line	130
8.12.12. Interrupt Pin	130
9. ISA INTERFACE	131
9.1. INTRODUCTION	131
9.2. PCI / ISA CYCLES	131
9.2.1. PCI to ISA read and write	131
9.2.2. PCI TO INTERNAL REGISTER READ AND WRITE	132
9.2.3. Interrupt Acknowledge Cycle	132
9.2.4. ISA to PCI read and write	133
9.2.5. ISA to PCI buffered reads	133
9.2.6. ISA to PCI posted writes	133
9.2.7. ISA to register read and write	133
9.3. XBUS READ AND WRITE	134
9.3.1. Real Time Clock Read and Write	134
9.3.2. BIOS ROM read and write	134
9.3.3. CPU Reset and Gate A20	135
9.4. ISA STANDARD REGISTERS	136
9.4.1. DMA 1 controller registers	136
9.4.2. Interrupt controller 1 registers	137
9.4.3. Interval Timer registers	138
9.4.4. Port B register	139
9.4.5. Port 70h register	140
9.4.6. Interrupt Controller 2 registers	141

9.4.7. DMA Controller 2 registers	142
9.4.8. DMA Page registers	143
9.5. ISA CONFIGURATION REGISTERS	144
9.5.1. Miscellaneous Control Register 0	144
9.5.2. Miscellaneous Control register 1	145
9.5.3. PIRQ Routing control register 0	146
9.5.4. Interrupt Level Control Register 0	147
9.5.5. Interrupt Level Control Register 1	148
9.5.6. IPC Configuration register	149
9.6. INTERRUPT ROUTER	151
9.6.1. INTERRUPT CONTROLLER REGISTER SUMMARY	152
9.6.2. INTERRUPT Routing control registers	153
9.7. DRQ ROUTER	154
9.7.1. DRQ Interrupt Router Summary	155
9.7.2. DRQ Routing control registers	156
10. UIDE CONTROLLER	157
10.1.INTRODUCTION	157
10.2.ATA4 COMPLIANT UIDE CONTROLLER	157
10.3.UDMA CONTROL REGISTER	158
10.4.OPERATION	159
10.4.1.Signal Descriptions	159
10.5.IDE CONTROLLER OPERATION	161
10.6.PRD TABLE ENTRY	162
10.7.IDE BUS MASTER REGISTERS	163
10.7.1.Physical Region Descriptor Table	163
10.7.2.Physical Region Descriptor	164
10.8.BUS MASTER IDE REGISTER DESCRIPTION	164
10.9.BUS MASTER IDE COMMAND REGISTER	165
10.9.1.IDE Command Register	165
10.9.2.IDE Status Register	166
10.9.3.Descriptor Table Pointer Register	168
10.10.OPERATION	169
10.10.1.Standard Programming Sequence	169
10.11.DATA SYNCHRONIZATION	169
10.11.1.Status Bit Interpretation	170

10.12.ERROR CONDITIONS	170
10.13.PCI SPECIFICS	170
11. LOCAL BUS INTERFACE	173
11.1.INTRODUCTION	173
11.1.1.Features	173
11.2.MEMORY BANK SWITCHING	174
11.3.FLASH DEVICE IMPLEMENTATION	174
11.3.1.Standard BIOS Boot or Boot Loader in Real Mode	174
11.3.2.Bootloader that is executed above the first MByte	176
11.4.CONFIGURATION REGISTERS	178
11.5.LOCAL BUS BASE INDEX REGISTER	180
11.5.1.InitialiSation	180
11.5.2.write access	180
11.5.3.Read access	180
11.5.4.I/O slot base address registers IOAREG0 to IOAREG7	181
11.5.5.I/o slot mask registerS	182
11.5.6.memory base address register 0	183
11.5.7.memory base address register 1	184
11.5.8.MEMory MASK register	185
11.6.LOCAL BUS TIMING REGISTERS	186
11.6.1.TIMING MEMORY TEMPLATE REGISTER 0	186
11.6.2.TIMING MEMORY TEMPLATE REGISTER 1	187
11.6.3.i/o timing template register 0	188
11.6.4.i/o timing template register 1	189
11.6.5.i/o timing template register 2	190
11.6.6.i/o timing template register 3	191
11.6.7.i/o timing template register 4	192
11.6.8.i/o timing template register 5	193
11.6.9.i/o timing template register 6	194
11.6.10.i/o timing template register 7	195
11.7.LOCAL BUS CONTROL REGISTER	196
11.8.LOCAL BUS DEVICE WIDTH REGISTER	198
12. GPIO Interface	199
12.1.INTRODUCTION	199
12.2.GPIO CONFIGURATION	199
12.3.REGISTER DESCRIPTION	200

12.3.1.Port Direction Control Register (Base+00h):	200
12.3.2.Read Port Control Register (Base+01h):	201
12.3.3.Read register (Base+02h):	202
12.3.4.INTERRUPT UNMASK REGISTER (Base+03h):	203
12.3.5.INTERRUPT EDGE REGISTER (Base+04h):	204
12.3.6.INTERRUPT CLEAR COMMAND(Base+05h):	205
12.3.7.GPIO port register (Base+06h):	206
12.3.8.STRAP REGISTER (Base+07h):	207
13. UNIVERSAL SERIAL BUS	209
13.1.INTRODUCTION	209
13.2.OPERATIONAL REGISTERS	210
13.3.PCI CONFIGURATION	213
13.3.1.PCI INTERFACE	213
13.3.2.PCI Configuration Spaces for OpenHCI-compliant USB Host Controller ..	214
14. MAC ETHERNET INTERFACE (LAN)	215
14.1.INTRODUCTION	215
14.2.FEATURES	215
14.3.FUNCTIONAL DESCRIPTION	216
14.3.1.LAN Target Block	216
14.3.2.LAN Master Block	216
14.3.3.MAC Block	217
14.3.4.PHY Block	217
14.4.I/O SPACE REGISTER DESCRIPTION	218
14.5.REGISTER BIT FIELD DESCRIPTIONS	220
14.5.1.Bus Mode Register	220
14.5.2.Transmit Poll Demand Register	221
14.5.3.Receive Poll Demand Register	222
14.5.4.Receive Descriptor Ring Base Address	223
14.5.5.Transmit Descriptor Ring Base Address	224
14.5.6.Status Register	225
14.5.7.Interrupt Enable Register	228
14.5.8.Current Tx Descriptor Pointer	230
14.5.9.Current Rx Descriptor Pointer	231
14.5.10.MAC Control Register	232
14.5.11.MAC Address Hi and Lo Registers	236
14.5.12.Multicast Address Hi/Lo RegisterS	238
14.5.13.MII Address Register	239

14.5.14.MII Data Register	240
14.5.15.Flow Control Register	241
14.5.16.VLAN1 Tag Register	243
14.5.17.VLAN2 Tag Register	244
14.6.HOST COMMUNICATION	245
14.7.DESRIPTOR LISTS AND DATA BUFFERS	245
14.7.1.Receive Descriptors (RDES)	245
14.7.2.Transmit Descriptors	252
15. SERIAL PORT	259
15.1.INTRODUCTION	259
15.2.FUNCTIONAL DESCRIPTION	259
15.2.1.Transmit Operation	259
15.2.2.Receive Operation	259
15.2.3.Modem Control Lines	259
15.3.SERIAL INTERFACE SIGNALS	259
15.4.REGISTER DESCRIPTION	260
15.4.1.Addressing	260
15.4.2.Receiver Buffer Register	261
15.4.3.Transmitter Holding Register	262
15.4.4.Interrupt Enable Register	263
15.4.5.Interrupt Identification Register	264
15.4.6.Receive Timeout Interrupt	265
15.4.7.TX FIFO Interrupt	265
15.4.8.FIFO Polled Operation	265
15.4.9.FIFO Control Register	266
15.4.10.Line Control Register	267
15.4.11.Modem Control Register	268
15.4.12.Line Status Register	269
15.4.13.Modem Status Register	271
15.4.14.Scratch Register	272
15.4.15.Divisor Latch (LS) - divisor latch (ms)	273
15.5.SPECIAL FEATURES	274
15.5.1.Transmit Machine Timing	274
15.5.2.THR Empty Interrupt Timing	274
15.5.3.FIFO Reset Timing	274
16. I2C BUS CONTROLLER	275
16.1.INTRODUCTION	275

16.2.FEATURES	275
16.3.FUNCTIONAL DESCRIPTION	276
16.4.I2C BUS OPERATION	276
16.5.SERIAL CLOCK LINE (SCL) GENERATION	277
16.6.I2C CONTROLLER MASTER OPERATION	277
16.7.I2C CONTROLLER SLAVE OPERATION	277
16.8.START AND STOP BUS STATES	278
16.9.GLITCH SUPPRESSION LOGIC	278
16.10.RESET CONDITIONS	279
16.11.MASTER OPERATION	279
16.12.SLAVE OPERATION	282
16.13.SLAVE MODE PROGRAMMING EXAMPLES	283
16.13.1.Initialize Controller	283
16.13.2.Write One byte as a Slave	283
16.13.3.Read Two Bytes as a Slave	284
16.14.MASTER PROGRAMMING EXAMPLES	284
16.14.1.Initialize Controller	284
16.14.2.Write One Byte as a Master	284
16.14.3.Read One Byte as a Master	284
16.15.CONFIGURATION REGISTERS	285
16.16.I2C CONTROLLER CONTROL REGISTERS	285
16.16.1.Buffer Receive Register	285
16.16.2.Buffer Transmit Register	286
16.16.3.Status Register	287
16.16.4.Command Register	289
16.16.5.Control Register	291
16.16.6.Clock Control Register	293
16.16.7.Scratch Register	294
16.16.8.Byte Count Register	295
17. POWER MANAGEMENT	297
17.1.INTRODUCTION	297
17.2.POWER MANAGEMENT CONTROLLER REGISTERS	299
17.2.1.TIMER REGISTER 0	299
17.2.2.TIMER REGISTER 1	301



17.2.3.TIMER REGISTER 2	303
17.2.4.SYSTEM ACTIVITY ENABLE REGISTER 0	304
17.2.5.SYSTEM ACTIVITY ENABLE REGISTER 1	305
17.2.6.SYSTEM ACTIVITY ENABLE REGISTER 2	306
17.2.7.HOUSE-KEEPING ACTIVITY ENABLE REGISTER 0	307
17.2.8.HOUSE-KEEPING ACTIVITY ENABLE REGISTER 1	308
17.2.9.PERIPHERAL INACTIVITY DETECTION REGISTER 0	309
17.2.10.PERIPHERAL ACTIVITY DETECTION REGISTER 0	310
17.2.11.PERIPHERAL ACTIVITY DETECTION REGISTER 1	311
17.2.12.ADDRESS RANGE 0 REGISTER 0	312
17.2.13.ADDRESS RANGE 0 REGISTER 1	313
17.2.14.SMI CONTROL REGISTER 0	314
17.2.15.SMI STATUS REGISTER 0	315
17.2.16.SMI STATUS REGISTER 1	317
17.2.17.PERIPHERAL INACTIVITY STATUS REGISTER 0	318
17.2.18.ACTIVITY STATUS REGISTER 0	319
17.2.19.ACTIVITY STATUS REGISTER 1	320
17.2.20.ACTIVITY STATUS REGISTER 2	321
17.2.21.PMU STATUS REGISTER	322
17.2.22.GENERAL PURPOSE REGISTER	324
17.2.23.CLOCK CONTROL REGISTER 0	325
17.2.24.DOZE TIMER READ BACK REGISTER	327
17.2.25.STANDBY TIMER READ BACK REGISTER	328
17.2.26.SUSPEND TIMER READ BACK REGISTER	329
17.2.27.HOUSE-KEEPING TIMER READ BACK REGISTER	330
17.2.28.PERIPHERAL TIMER READ BACK REGISTER	331

2 List of Tables

Table 8-1. Register CF8h	.72
Table 8-2. Register CFCh	.72
Table 8-3. North Bridge Reset Values	.74
Table 8-4. ISA Bridge Configuration Space Register Reset Values	.82
Table 8-5. PCI to IDE Bridge Configuration Space Register Reset Values	.89
Table 8-6. Operating Mode of the Secondary Channel	.93
Table 8-7. Operating Mode of the Primary Channel	.93
Table 8-8. PCI to USB Bridge Configuration Space Register Reset Values	.109
Table 8-9. PCI to LAN Bridge Configuration Space Register Reset Values	.123
Table 17-1. Activity Detected	.298
Table 17-2. Suspend Timer Reset	.299
Table 17-3. Standby Timer Reset	.300
Table 17-4. House-keeping Timer Reset	.301
Table 17-5. Peripheral Timer Reset	.302
Table 17-6. Doze Timer Reset	.303
Table 17-7. PMU State	.322
Table 17-8. Power-on and Housekeeping States	.325
Table 17-9. Doze/Standby/Suspend States	.326



3 List of Figures

Figure 6-1. STPC Host Layout	33
Figure 6-2. STPC Physical Memory Map	34
Figure 7-1. Memory Controller Interface Block Diagram	63
Figure 8-1. PCI Layout	71
Figure 9-1. Interrupt Router Schematic Layout	151
Figure 9-2. Interrupt Router Schematic Layout	154
Figure 10-1. PRD Table Entry Example	163
Figure 11-1. Standars BIOS Boot Illustration	175
Figure 11-2. Two flash device implimentation	177
Figure 14-1. Block Diagram: MAC Ethernet Controller	216



4. HOW TO USE THIS MANUAL

4.1. INTRODUCTION

This manual provides full technical documentation for the STPC device. It is recommended that the reader is familiar with the x86 series processors and PC compatible architectures before reading this document. Many terms are related directly to the PC architecture.

The manual itself is split into chapters. These chapters hold the information for a particular functional block of the device. For example, the chapter titled "Memory Access" gives the memory map of the STPC device, the memory architecture and interface to the external DRAM modules.

4.2. SPECIFIC NOTES

4.2.1. RESERVED BITS

Write mode 1 is a subset of Write Mode 0. No CPU-supplied write data is used. The read data latched from a previous read operation is written. The bit mask is disabled. The map-masks are implemented as they are for Write Mode 0.

Many bits in the register descriptions are noted as reserved. These bits are not internally connected, physically not present or are used for testing purposes. In all cases these bits should be set to a '0' when writing to a register with reserved bits. When reading from a register with reserved bits, these specific bits should be masked from the data value before action is taken on the data.

Any functionality found by setting the reserved bits to levels other than '0' cannot and will not be guaranteed on future revisions of the circuit design. Thus it is not recommended to use the bits marked as reserved in any way different from noted above.

4.2.2. SIGNAL ACTIVE STATE

The hash symbol (#) following a signal name indicates that when the signal is in its active (asserted) state, the signal is at a logic low level. When the "#" is not present at the end of a signal name, the logic high level represents the active state.

4.2.3. HEXADECIMAL NOTATION

In this manual Hexadecimal (Hex) numbers (numbers to the base 16: [0-9,A-F]) are denoted by the postfix 'h'.

For example a memory address 783A hexadecimal will be written 783Ah.

4.2.4. ENDIAN

In common with the x86 architecture, values in memory are little-endian, that is the lower part of the memory contains the least significant Byte.

For an 8-bit value

N	7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---	---

For a 16-bit (word) value

N	7	6	5	4	3	2	1	0
N+1	15	14	13	12	11	10	9	8

For a 24-bit value

N	7	6	5	4	3	2	1	0
N+1	15	14	13	12	11	10	9	8
N+2	23	22	21	20	19	18	17	16

For a 32-bit (long word) value

N	7	6	5	4	3	2	1	0
N+1	15	14	13	12	11	10	9	8
N+2	23	22	21	20	19	18	17	16
N+3	31	30	29	28	27	26	25	24

For a 64-bit (QUAD word) value

N	7	6	5	4	3	2	1	0
N+1	15	14	13	12	11	10	9	8
N+2	23	22	21	20	19	18	17	16
N+3	31	30	29	28	27	26	25	24
N+4	39	38	37	36	35	34	33	32
N+5	47	46	45	44	43	42	41	40
N+6	55	54	53	52	51	50	49	48
N+7	63	62	61	60	59	58	57	56



4.3. ISSUING NOTES

There are three levels identified; Advanced data, Preliminary data and Full production release.

Each level is identified in a specific way as follows.

Document Identification	Status	Definition	Release Identification
ADVANCED DATA	In design	This document based on the product specification. The information may be updated without notice. Large changes may still occur.	Release A, Release B...
PRELIMINARY DATA	Pre-production Data	This document contains preliminary data and may be updated without notice in order to improve the product features.	Issue 0.X.
FULL PRODUCTION DATA	Production Data	This is the finalised document and all test plans are completed. The information may be updated without notice in order to improve the product features.	Issue 1.X.

5 LIST OF REGISTERS

This chapter lists all the registers accessible by external software.

Section	Register Name	Mnemonic	Purpose	Address	Access type
3.1	Power on Strap Registers			0022/23h	
3.1.1	ADPC Strap Register 0	ADPC0	Configuration		Index 04Ah
3.1.2	ADPC Strap Register 1	ADPC1	Configuration		Index 04Bh
3.1.3	ADPC Strap Register 2	ADPC2	Configuration		Index 04Ch
6.5.	Cache Related Registers			0022h	
6.5.1.	Cache Architecture Register 0	Cash_arc0	Configuration	0023h	Index 020h
6.5.2.	Cache Architecture Register 1	Cash_arc1	Configuration		Index 021h
6.5.3.	Cache Architecture Register 2	Cash_arc2	Configuration		Index 022h
6.6.	Address Decode Related Registers			0022h	
6.6.1.	Memory Hole Control Register	MEM_HOLE	Configuration	0023h	Index 024h
6.6.2.	Shadow Control Register 0	SHADOW_0	Configuration		Index 025h
6.6.3.	Shadow Control Register 1	SHADOW_1	Configuration		Index 026h
6.6.4.	Shadow Control Register 2	SHADOW_2	Configuration		Index 027h
6.6.5.	Shadow Control Register 3	SHADOW_3	Configuration		Index 028h
6.7.	Host SDRAM Controller Registers			0022h	
6.7.1.	SDRAM Bank 0 Register	SDRAM_bank0	Configuration	0023h	Index 030h
6.7.2.	SDRAM Bank 1 Register	SDRAM_bank1	Configuration		Index 031h
6.7.3.	SDRAM Bank 2 Register	SDRAM_bank2	Configuration		Index 032h
6.7.4.	SDRAM Bank 3 Register	SDRAM_bank3	Configuration		Index 033h
6.7.5.	SDRAM Refresh Register	SDRAM_Ref	Configuration		Index 039h
7.3.	Memory Interface			GBase+4C6000h	
7.3.1.	Register 0	MEM_REG0	Configuration		000h
7.3.2.	Register 1	MEM_REG1	Configuration		004h
7.3.3.	Register 2	MEM_REG2	Configuration		008h
7.4.	MCLK Control Registers			22h	

LIST OF REGISTERS

Section	Register Name	Mnemonic	Purpose	Address	Access type
7.4.1.	MCLK Control Register 0	MCLK00		23h	Index 0x40h
7.4.2.	MCLK Control Register 1	MCLK01			Index 0x41h
8.3.	PCI CONFIGURATION ADDRESS REGISTER	Config_address	IO	0xCF8h	
8.4.	CONFIGURATION DATA REGISTER	Config_data	IO	0xCFC - CFFh	
8.5.	NORTH BRIDGE CONFIGURATION REGISTERS				
8.5.1.	North Bridge PCI Command Register	NB_Com	PCI Config		Index 0x4h
8.5.2.	North Bridge PCI Status Register	NB_Stat	PCI Config		Index 0x6h
8.5.3.	North Bridge PCI Revision ID Register	NB_R_ID	PCI Config		Index 0x8h
8.5.4.	North Bridge Device Class Code Register	NB_C_Code	PCI Config		Index 0x9h
8.5.5.	North Bridge Header Type Register	NB_Hesd	PCI Config		Index 0x0Eh
8.5.6.	North Bridge Control Register	NB_Cont	PCI Config		Index 0x50h
8.5.7.	North Bridge PCI Error Status Register	NB_E_Stat	PCI Config		Index 0x54h
8.7.	SOUTH BRIDGE PCI ISA CONFIGURATION REGISTERS			0xCF8h	
8.7.1.	South Bridge PCI Command Register	SB_Com0	PCI Config F#0		Index 0x4h
8.7.2.	South Bridge PCI Status Register	SB_Stat0	PCI Config F#0		Index 0x6h
8.7.3.	South Bridge PCI Revision ID Register	SB_R_ID0	PCI Config F#0		Index 0x8h
8.7.4.	South Bridge Device Class Code Register	SB_C_Code0	PCI Config F#0		Index 0x9h
8.7.5.	South Bridge Header Type Register	SB_Head0	PCI Config F#0		Index 0xEh
8.7.6.	South Bridge Miscellaneous Register	SB_Misc0			Index 040h
8.8.	PCI to IDE BRIDGE CONFIGURATION REGISTERS				
8.8.1.	PCI to IDE Bridge PCI Command Register	IDEB_Com1	PCI Config F#1		Index 0x4h

Section	Register Name	Mnemonic	Purpose	Address	Access type
8.8.2.	PCI to IDE Bridge PCI Status Register	IDEB_Stat1	PCI Config F#1		Index 0x6h
8.8.3.	PCI to IDE Bridge Revision ID Register	IDEB_R_ID1	PCI Config F#1		Index 0x8h
8.8.4.	PCI to IDE Bridge Programming Interface Register	Prog_Int	PCI Config F#1		Index 0x9h
8.8.5.	PCI to IDE Bridge Sub-Class Code Register	Sub_Class	PCI Config F#1		Index 0xAh
8.8.6.	PCI to IDE Bridge Base-Class code Register	Base_Class	PCI Config F#1		Index 0xBh
8.8.7.	PCI to IDE Bridge Latency Timer control Register	Lat_T	PCI Config F#1		Index 0xDh
8.8.8.	PCI to IDE Bridge Header Type Register	Head_T	PCI Config F#1		Index 0xEh
8.8.9.	PCI to IDE Bridge Base Address 0 Register	Base0	PCI Config F#1		Index 0x10h
8.8.10.	PCI to IDE Bridge Base Address 1 Register	Base1	PCI Config F#1		Index 0x14h
8.8.11.	PCI to IDE Bridge Base Address 2 Register	Base2	PCI Config F#1		Index 0x18h
8.8.12.	PCI to IDE Bridge Base Address 3 Register	Base3	PCI Config F#1		Index 0x1Ch
8.8.13.	PCI to IDE Bridge Base Address 4 Register	Base4	PCI Config F#1		Index 0x20h
8.8.14.	Misc Register 0	MISC_REG0			Index 0x40h
8.8.15.	Misc Register 1	MISC_REG1			Index 0x48h
8.8.16.	Primary Master Timing Register	PMT_REG			Index 0x50h
8.8.17.	Primary Slave Timing Register	PST_REG			Index 0x54h
8.8.18.	Secondary Master Timing Register	SMT_REG			Index 0x58h
8.8.19.	Secondary Slave Timing Register	SST_REG			Index 0x5Ch
8.9.	PCI to USB BRIDGE CONFIGURATION REGISTERS			0xCF8h/ 0xCFCh	
8.9.1.	PCI to USB Bridge Vendor Identification Register	USBB_V_ID1			Index 0x00h
8.9.2.	PCI to USB Bridge Device Identification Register	USBB_D_ID1			Index 0x02h
8.9.3.	USB Bridge PCI Command Register	USBB_Com			Index 0x04h
8.9.4.	USB Bridge PCI Status Register	USBB_Stat			Index 0x06h

LIST OF REGISTERS

Section	Register Name	Mnemonic	Purpose	Address	Access type
8.9.5.	USB Bridge PCI Revision ID Register	USBB_R_ID			Index 0x08h
8.9.6.	USB Bridge Device Class Code Register	USBB_C_Code			Index 0x09h, 0Ah & 0Bh
8.9.7.	USB Bridge Header Type Register	USBB_Head			Index 0x0Eh
8.10.	PCI Configuration for OpenHCI-compliant USB Host Controller				
8.10.1.	Command Register	Command			Index 0x h
8.10.2.	Class Code	CLASS_CODE			Index 0x h
8.10.3.	Base Address Register	BAR_OHCI			Index 0x h
8.12.	PCI to LAN Bridge Configuration Registers			0xCF8h/ 0xCFCh	
8.12.1.	Command Register	COMMAND			05h
8.12.2.	Status Register	STATUS			06h
8.12.3.	Revision ID Register	REVISION_ID			08h
8.12.4.	Class Code Register	CLASS_CODE			09h
8.12.5.	Cache Line Size Register	CACHE_LINE_SIZE			0Ch
8.12.6.	Latency Timer Register	LATENCY_TIMER			0Dh
8.12.7.	Header Type Register	HEADER_TYPE			0Eh
8.12.8.	BIST Register	BIST			0Fh
8.12.9.	Memory Base Address Register 0	MEMORY_BASE_ADDRESS			10h
8.12.10.	I/O Base Address Register 0	IO_BASE_ADDRESS			14h
8.12.11.	Interrupt Line Register	INTERRUPT_LINE			3Ch
8.12.12.	Interrupt Pin Register	INTERRUPT_PIN			3Dh
9.4.	ISA Standard Registers				
9.4.1.	DMA 1 Controller Registers	DMA_1	IO	0000h	
9.4.1.	DMA 1 Channel 0 Base and Current Count	DMA1_CBA0	IO	0001h	
9.4.1.	DMA 1 Channel 1 Base and Current Address	DMA1_CBC0	IO	0002h	

Section	Register Name	Mnemonic	Purpose	Address	Access type
9.4.1.	DMA 1 Channel 1 Base and Current Count	DMA1_CBA1	IO	0003h	
9.4.1.	DMA 1 Channel 2 Base and Current Address	DMA1_CBC1	IO	0004h	
9.4.1.	DMA 1 Channel 2 Base and Current Count	DMA1_CBA2	IO	0005	
9.4.1.	DMA 1 Channel 3 Base and Current Address	DMA1_CBC2	IO	0006h	
9.4.1.	DMA 1 Channel 3 Base and Current Count	DMA1_CBA3	IO	0007h	
9.4.1.	DMA 1 Read Status / Write Command Register	DMA1_RSWC	IO	0008h	
9.4.1.	DMA 1 Request Register	DMA1_RR	IO	0009h	
9.4.1.	DMA 1 Read Command / Write Single Mask Register	DMA1_RCWSM	IO	000Ah	
9.4.1.	DMA 1 Mode Register	DMA1_Mode	IO	000Bh	
9.4.1.	DMA 1 Set / Clear Byte Pointer Flip - Flop	DMA1_SCBPFF	IO	000Ch	
9.4.1.	DMA 1 Read Temp Register / Master Clear	DMA1_RTMC	IO	000Dh	
9.4.1.	DMA 1 Clear Mask / Clear All Request	DMA1_CMCAR	IO	000Eh	
9.4.1.	DMA 1 Read / Write all Mask Register Bits	DMA1_RWMB	IO	000Fh	
9.4.2.	Interrupt Controller 1 Registers	IC_1	IO	0020h	
9.4.3.	Interval Timer Registers	IT_1	IO	0040h	
9.4.3.	Interval Timer Register Counter 0 Count	IT_0	IO	0040h	
9.4.3.	Interval Timer Register Counter 1 Count	IT_1	IO	0041h	
9.4.3.	Interval Timer Register Counter 2 Count	IT_2	IO	0042h	
9.4.3.	Command Mode Register	IT_3	IO	0043h	
9.4.4.	Port B Register	Port_B	IO	0061h	
9.4.5.	Port 60h Register	Port_60	IO	0060h	
9.4.5.	Port 64h Register	Port_64	IO	0064h	
9.4.5.	Port 70 Register	Port_70	IO	0070h	
9.4.6.	Interrupt Controller 2 Registers	IC_2	IO	00A0h	
9.4.7.	DMA Controller 2 Registers	DMA_Cont2	IO		

LIST OF REGISTERS

Section	Register Name	Mnemonic	Purpose	Address	Access type
9.4.7.	DMA 2 Channel 0 Base and Current Address	DMA2_CBA0	IO	00C0h	
9.4.7.	DMA2 Channel 0 Base and Current Count	DMA2_CBC0	IO	00C2h	
9.4.7.	DMA 2 Channel 1 Base and Current Address	DMA2_CBA1	IO	00C4h	
9.4.7.	DMA 2 Channel 1 Base and Current Count	DMA2_CBC1	IO	00C6h	
9.4.7.	DMA 2 Channel 2 Base and Current Address	DMA2_CBA2	IO	00C8h	
9.4.7.	DMA 2 Channel 2 Base and Current Count	DMA2_CBC2	IO	00CAh	
9.4.7.	DMA 2 Channel 3 Base and Current Address	DMA2_CBA3	IO	00CCh	
9.4.7.	DMA 2 Channel 3 Base and Current Count	DMA2_CBC3	IO	00CEh	
9.4.7.	DMA 2 Read Status / Write Command Register	DMA2_RSWC	IO	00D0h	
9.4.7.	DMA 2 Request Register	DMA2_RR	IO	00D2h	
9.4.7.	DMA 2 Read Command / Write Single Mask Register	DMA2_RCWSM	IO	00D4h	
9.4.7.	DMA 2 Mode Register	DMA2_Mode	IO	00D6h	
9.4.7.	DMA 2 Set / Clear Byte Pointer Flip - Flop	DMA2_SCBPFF	IO	00D8h	
9.4.7.	DMA 2 Read Temporary / Master Clear	DMA2_RTMC	IO	00DAh	
9.4.7.	DMA 2 Clear Mask / Clear All Requests Register	DMA2_CMCAR	IO	00DCh	
9.4.7.	DMA 2 Read / Write all Mask Register Bits	DMA2_RWMRB	IO	00DEh	
9.4.8.	DMA Page Registers	DMA_Page	IO		
9.4.8.	DMA Page Registers Port 80h (reserved)	Port_80	IO	0080h	
9.4.8.	DMA Page Register Channel 2	DMA_PRC2	IO	0081h	
9.4.8.	DMA Page Register Channel 3	DMA_PRC3	IO	0082h	
9.4.8.	DMA Page Register Channel 1	DMA_PRC1	IO	0082h	
9.4.8.	DMA Page Register Port 84h	Port_84	IO	0084h	(Reserved)
9.4.8.	DMA Page Register Port 85h	Port_85	IO	0085h	(Reserved)
9.4.8.	DMA Page Register Port 86h	Port_86	IO	0086h	(Reserved)
9.4.8.	DMA Page Register Channel 0	DMA_PRC0	IO	0087h	

Section	Register Name	Mnemonic	Purpose	Address	Access type
9.4.8.	DMA Page Register Port 87h	Port_87	IO	0088h	
9.4.8.	DMA Page Register Channel 6	DMA_PRC6	IO	0089h	
9.4.8.	DMA Page Register Channel 7	DMA_PRC7	IO	008Ah	
9.4.8.	DMA Page Register Channel 5	DMA_PRC5	IO	008Bh	
9.4.8.	DMA Page Register Port 8Bh	Port_8B	IO	008Ch	(Reserved)
9.4.8.	DMA Page Register Port 8Ch	Port_8C	IO	008Dh	(Reserved)
9.4.8.	DMA Page Register Port 8Dh	Port_8D	IO	008Eh	(Reserved)
9.4.8.	DMA Page Register Port 8Eh	Port_8E	IO	008Fh	(Reserved)
9.5.	ISA Configuration Registers			0022h	
9.5.1.	Miscellaneous Control Register 0	Misc_Cont0	Configuration	0023h	Index 050h
9.5.2.	Miscellaneous Control Register 1	Misc_Cont1	Configuration		Index 051h
9.5.3.	PIRQA Routing control Register 0	PAR_Cont0	Configuration		Index 052h
9.5.3.	PIRQB Routing control Register 0	PBR_Cont0	Configuration		Index 053h
9.5.3.	PIRQC Routing control Register 0	PCR_Cont0	Configuration		Index 054h
9.5.3.	PIRQD Routing control Register 0	PDR_Cont0	Configuration		Index 055h
9.5.4.	Interrupt Level Control Register 0	IRQ_Lev_C_0	Configuration		Index 056h
9.5.5.	Interrupt Level Control Register 1	IRQ_Lev_C_1	Configuration		Index 057h
9.5.6.	IPC Configuration Register	IPC_Conf	Configuration		Index 001h
10.	UIDE Controller			See UIDE Controller Chapter	
11.	Local Bus Registers		16 bit access	22h	
	Local Bus Address Decode Registers			23h	
11.5.4.	I/O Slot Base Register 0	IOAREG0	Base address		Index 0x0000h
11.5.4.	I/O Slot Base Register 1	IOAREG1	Base address		Index 0x0002h
11.5.4.	I/O Slot Base Register 2	IOAREG2	Base address		Index 0x0004h
11.5.4.	I/O Slot Base Register 3	IOAREG3	Base address		Index 0x0006h
11.5.4.	I/O Slot Base Register 4	IOAREG4	Base address		Index 0x0008h
11.5.4.	I/O Slot Base Register 5	IOAREG5	Base address		Index 0x000Ah
11.5.4.	I/O Slot Base Register 6	IOAREG6	Base address		Index 0x000Ch
11.5.4.	I/O Slot Base Register 7	IOAREG7	Base address		Index 0x000Eh
11.5.5.	I/O Slot Mask Register 0	IOMREG0	Mask size		Index 0x0010h
11.5.5.	I/O Slot Mask Register 1	IOMREG1	Mask size		Index 0x0012h
11.5.5.	I/O Slot Mask Register 2	IOMREG2	Mask size		Index 0x0014h



LIST OF REGISTERS

Section	Register Name	Mnemonic	Purpose	Address	Access type
11.5.5.	I/O Slot Mask Register 3	IOMREG3	Mask size		Index 0x0016h
11.5.5.	I/O Slot Mask Register 4	IOMREG4	Mask size		Index 0x0018h
11.5.5.	I/O Slot Mask Register 5	IOMREG5	Mask size		Index 0x001Ah
11.5.5.	I/O Slot Mask Register 6	IOMREG6	Mask size		Index 0x001Ch
11.5.5.	I/O Slot Mask Register 7	IOMREG7	Mask size		Index 0x001Eh
11.5.6.	Base address Memory Bank0	MEMAREG0			Index 0x0FE0h
11.5.7.	Base address Memory Bank1	MEMAREG1			Index 0x0FC0h
11.5.8.	Address Range Mem Bank0 & 1	MEMMASK			Index 0x003Fh
	Local Bus Timing Registers				
11.6.1.	Memory Timing Template 0	TIMEBANK0	Access Bank0		Index 0x0020h
11.6.2.	Memory Timing Template 1	TIMEBANK1	Access Bank1		Index 0x0022h
11.6.3.	I/O Timing Template 0	TIMEIO0	Access I/O 0		Index 0x0024h
11.6.4.	I/O Timing Template 1	TIMEIO1	Access I/O 1		Index 0x0026h
11.6.5.	I/O Timing Template 2	TIMEIO2	Access I/O 2		Index 0x0028h
11.6.6.	I/O Timing Template 3	TIMEIO3	Access I/O 3		Index 0x002Ah
11.6.7.	I/O Timing Template 4	TIMEIO4	Access I/O 4		Index 0x002Ch
11.6.8.	I/O Timing Template 5	TIMEIO5	Access I/O 5		Index 0x002Eh
11.6.9.	I/O Timing Template 6	TIMEIO6	Access I/O 6		Index 0x0030h
11.6.10.	I/O Timing Template 7	TIMEIO7	Access I/O 7		Index 0x0032h
	Local Bus Control Register				
11.7.	Control Register	CONTROL			Index 0x0001h
11.8.	I/O or Mem Width Register	IOWIDTH			Index 0x0300h
12.	GPIO				
12.3.1.	Port Direction Control Register	portDirCtrl			Index 0x0000h
12.3.2.	Read Port Control Register	readPortCtrl			Index 0x0001h
12.3.3.	Read Register	readReg			Index 0x0002h
12.3.4.	Interrupt Unmask Register	intrUnMask			Index 0x0011h
12.3.5.	Interrupt Edge Register	intrEdgeSelect			Index 0x0100h
12.3.6.	Interrupt Clear Command	clearIntr			Index 0x0101h
12.3.7.	GPIO Port Register	GPIOport			Index 0x0110h
12.3.8.	Strap Register	strapReg			Index 0x0111h

Section	Register Name	Mnemonic	Purpose	Address	Access type
15.	Serial Port			03F8h/ 02F8h	
15.4.2.	Receiver Buffer Register	RBR			Index 0x00h
15.4.3.	Transmitter Holding Register	THR			Index 0x00h
15.4.4.	Interrupt Enable Register	IER			Index 0x01h
15.4.5.	Interrupt Identification Register	IIR			Index 0x02h
15.4.9.	FIFO Control Register	FCR			Index 0x02h
15.4.10.	Line Control Register	LCR			Index 0x03h
15.4.11.	Modem Control Register	MCR			Index 0x04h
15.4.12.	Line Status Register	LSR			Index 0x05h
15.4.13.	Modem Status Register	MSR			Index 0x06h
15.4.14.	Scratch Register				Index 0x07h
15.4.15.	Divisor Latch (LS) Register	DLL			Index 0x00h
15.4.15.	Divisor Latch (MS) Register	DLM			Index 0x01h
14.	MAC Ethernet Interface (LAN)				
14.5.1.	Bus Mode Register	BUS_MODE			03h
14.5.2.	Transmit Poll Demand Register	TRANSMIT_POLL_DEMAND			04h
14.5.3.	Receive Poll Demand Register	RECEIVE_POLL_DEMAND			08h
14.5.4.	Rx Descriptor Ring Base Address Register	Rx_DESCR_BASE_ADD			0Fh
14.5.5.	Tx Descriptor Ring Base Address Register	Tx_DESCR_BASE_ADD			13h
14.5.6.	Status Register	STATUS			17h
14.5.7.	Interrupt Enable Register	INTERRUPT_ENABLE			1Eh
14.5.8.	Current Tx Descriptor Pointer Register	CURRENT_TX_DP			2Fh
14.5.9.	Current Rx Descriptor Pointer Register	CURRENT_RX_DP			33h
14.5.10.	MAC Control Register	MAC_CONTROL			21h
14.5.11.	MAC Address Low Register	MAC_ADDRESS_LO			37h
14.5.11.	MAC Address Hi Register	MAC_ADDRESS_HI			3Bh
14.5.12.	Multicast Address Low Register	MULTICAST_ADDRESS_LO			3Fh



LIST OF REGISTERS

Section	Register Name	Mnemonic	Purpose	Address	Access type
14.5.12.	Multicast Address Hi Register	MULTICAST_A DDRESS_HI			43h
14.5.13.	MII Address Register	MII_ADDRESS			47h
14.5.14.	MII Data Register	MII_DATA			4Bh
14.5.15.	Flow Control Register	FLOW_CONTR OL			4Fh
14.5.16.	VLAN1 Tag Register	VLAN1_TAG			53h
14.5.17.	VLAN2 Tag Register	VLAN2_TAG			57h
	I ² C Interface				
16.16.1.	Buffer Receive Register	Buffer_Receive			Index 0x00h
16.16.2.	Buffer_Transmit Register	Buffer_Transmit			Index 0x00h
16.16.3.	Status Register	Status			Index 0x04h
16.16.4.	Command Register	Command			Index 0x08h
16.16.5.	Control Register	Control			Index 0x0Ch
16.16.6.	Clock Control Register	Clock_Control			Index 0x10h
16.16.7.	Scratch Register	Scratch			Index 0x14h
16.16.8.	Byte Count Register	Byte_Count			Index 0x18h
13.	USB Port			See USB Port Chapter	
17.2.	Power Management Controller Registers:			0022h	
17.2.1.	Timer Register 0	Timer0	Configuration	0023h	Index 060h
17.2.2.	Timer Register 1	Timer1	Configuration		Index 061h
17.2.3.	Timer Register 2	Timer2	Configuration		Index 08dh
17.2.4.	System Activity Enable Register 0	Sys_activ_en0	Configuration		Index 062h
17.2.5.	System Activity Enable Register 1	Sys_activ_en1	Configuration		Index 063h
17.2.6.	System Activity Enable Register 2	Sys_activ_en2	Configuration		Index 064h
17.2.7.	House-Keeping Activity Enable Register 0	HK_activ_en0	Configuration		Index 065h
17.2.8.	House-Keeping Activity Enable Register 1	HK_activ_en1	Configuration		Index 066h
17.2.9.	Peripheral Inactivity Detection Register 0	Perif_inactiv0	Configuration		Index 067h
17.2.10.	Peripheral Activity Detection Register 0	Perif_activ0	Configuration		Index 069h
17.2.11.	Peripheral Activity Detection Register 1	Perif_activ1	Configuration		Index 06Ah

Section	Register Name	Mnemonic	Purpose	Address	Access type
17.2.12.	Address Range 0 Register 0	Add_range0-0	Configuration		Index 06Bh
17.2.13.	Address Range 0 Register 1	Add_range0-1	Configuration		Index 06Ch
17.2.14.	SMI Control Register 0	SMI_cont0	Configuration		Index 071h
17.2.15.	SMI Status Register 0	SMI_stat0	Configuration		Index 073h
17.2.16.	SMI Status Register 1	SMI_stat1	Configuration		Index 074h
17.2.17.	Peripheral Inactivity Status Register 0	Perif_stat0	Configuration		Index 075h
17.2.18.	Activity Status Register 0	Activ_stat0	Configuration		Index 077h
17.2.19.	Activity Status Register 1	Activ_stat1	Configuration		Index 078h
17.2.20.	Activity Status Register 2	Activ_stat2	Configuration		Index 079h
17.2.21.	PMU State Register	PMU	Configuration		Index 07Ah
17.2.22.	General Purpose Register	GP	Configuration		Index 07Bh
17.2.23.	Clock Control Register 0	Clock_cont0	Configuration		Index 07Ch
17.2.24.	Doze Timer Read Back Register	Doze	Configuration		Index 088h
17.2.25.	Stand-by Timer Read Back Register	Stand-by	Configuration		Index 089h
17.2.26.	Suspend Timer Read Back Register	Suspend	Configuration		Index 08Ah
17.2.27.	House-Keeping Timer Read Back Register	HK_timer	Configuration		Index 08Bh
17.2.28.	Peripheral Timer Read Back Register	Perif_timer	Configuration		Index 08Ch
<i>Note 2: X is the value of the G_Base and is set to 18000000h.</i>					



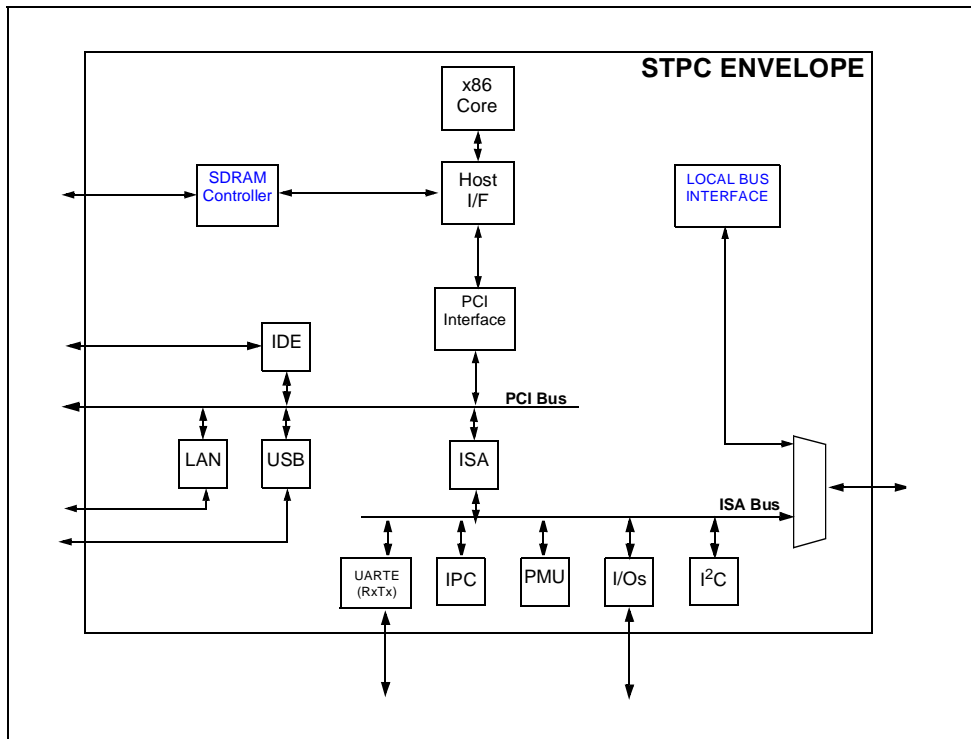
6. HOST INTERFACE

6.1. INTRODUCTION

This Chapter describes the Memory and I/O Mapping of the STPC with details on how to configure the Cache Memory registers.

The Host is the main interface between the CPU and the other integrated peripherals of the STPC. [Table 6-1](#) below illustrates the relation of the integrated devices with reference to the Host Interface.

Figure 6-1. STPC Host Layout



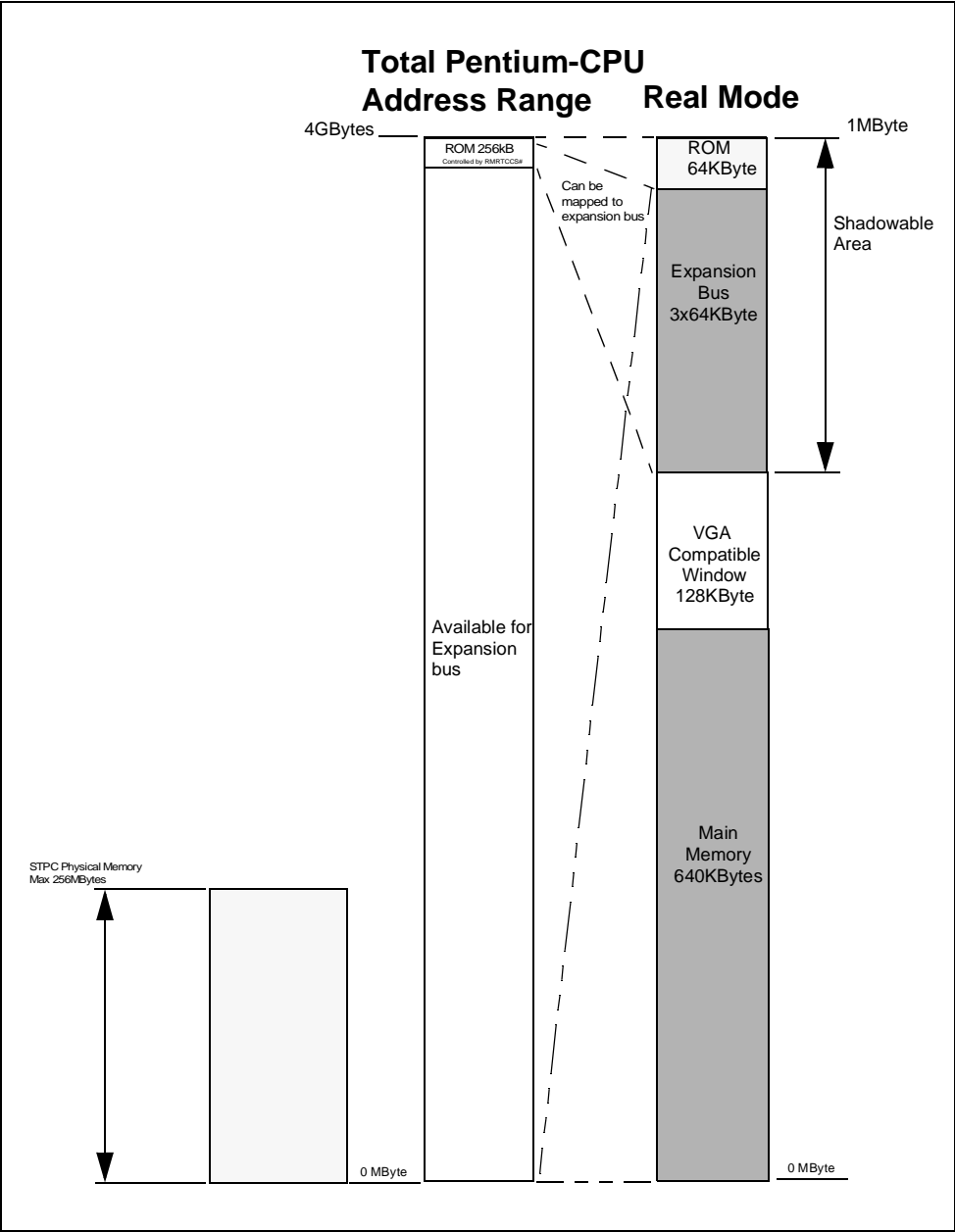


Figure 6-2. STPC Physical Memory Map

6.2. AGENT DECODING

All agents are decoded on a priority basis for instructions that are sent from the Host onto the host bus. If no agent on the Host Bus claims the cycle, it is then taken by the Host to PCI bridge (PCI North Bridge). If no agent on the PCI bridge claims the cycle it is forwarded to the ISA bridge.

For PCI Memory accesses, the cycle is forwarded to the Host Bridge to be decoded in by the SDRAM Controller. PCI Master cycles follow the procedure above. For ISA Master devices, the cycle is first forwarded to the PCI Bridge and follows the procedure described above. ISA Memory cycles are forwarded in the same way as the PCI Memory cycles.

6.3. MEMORY ADDRESS MAP

Figure 6-2 illustrate the STPC Memory Map including the general overview of how the SDRAM controller is situated within the complete map including the STPC Frame Buffer Location.

Memory Region	Address Range	Description
MAIN MEMORY (640K)	00000000h 0009FFFFh	<p>Host access maps to the main memory and no ISA or PCI cycle will be initiated. PCI master cycles in this range maps to main memory provided they are not claimed by a PCI Slave. The STPC relies on subtractive decode before initiating an internal memory cycle. ISA master cycles in this range maps to main memory. The STPC will negate IOCHRDY if necessary.</p> <p>The DMA master cycles in this range maps to main memory. The STPC will actively drive the SD bus during target reads and modify main memory for target write transfers.</p> <p>This address segment is considered always cacheable in the L1 cache. PCI and ISA master cycles in this range, require the L1 cache.</p>
VGA FRAME BUFFER (128K)	000A0000h 000BFFFFh	<p>This 128K address segment contains the VGA Frame buffer. Normally this address segment is mapped to the DOS frame buffer located in the main memory. However, if VGA is disabled or the VGA memory map mode is such that the VGA does not occupy the entire 128K address range, the host cycle is forwarded to the PCI bus and if not claimed by a PCI slave, it is further forwarded to the ISA bus.</p> <p>The PCI master cycles in this range, if not claimed by a PCI slave, will be mapped to the main memory or will be forwarded to the ISA bus as per the VGA decode described above.</p> <p>Similarly, the ISA or DMA master cycles will either map to the main memory or will be forwarded to the PCI. If no PCI slave claims the cycle, the STPC assumes existence of an ISA memory device at this address range.</p> <p>This segment is never cacheable.</p>
SHADOW (16K)	000C0000h 000C3FFFh	<p>This 16K address segment can be programmed software to either map to main memory or expansion buses. Further, reads and writes can have different mappings. If mapped to main memory, this segment will behave as the 0-640K segment.</p> <p>If not mapped to main memory, refer to Section 6.2. above</p> <p>If mapped to the main memory, the cacheability of this address range is controlled by software. If mapped to the ISA bus, the ROMCS# signal may optionally be asserted as controlled by software. This allows the system and video/peripheral BIOS to physically reside in a single ROM device.</p>

HOST INTERFACE

Memory Region	Address Range	Description
SHADOW (16K)	000C4000h 000C7FFFh	This range has the same characteristics as that of 000C0000h-000C3FFFh segment, as described above. The shadow control for this address range is provided via Shadow Control register 0 and cacheability and ROM chip-select control via Shadow Control register 3.
SHADOW (16K)	000C8000h 000CBFFFh	This range has the same characteristics as that of 000C0000h-000C3FFFh segment as described above, with the exception of the cacheability attribute. This address range is hardwired to be non-cacheable. Shadow control for this address range is provided via Shadow Control register 0 and ROM chip-select control via Shadow Control register 3.
SHADOW (16K)	000CC000h 000CFFFFh	This range has the same characteristics as that of 000C8000h-000CBFFFh segment, as described above. Shadow control for this address range is provided via Shadow Control register 0 and ROM chip-select control via Shadow Control register 3. This address range is hardwired to be non-cacheable.
SHADOW (64K)	000D0000h 000DFFFFh	This range has the same characteristics as that of 000CC000h-000CFFFFh segment, as described above. Shadow control for this address range is provided via Shadow Control register 1 and can be controlled at 16K resolution. ROM chip-select generation for the entire 64K range can be controlled via Shadow Control register 3. This address range is hardwired to be non-cacheable.
SHADOW (64K)	000E0000h 000EFFFFh	This range has the same characteristics as that of 000CC000h-000CFFFFh segment, as described above. Shadow control for this address range is provided via Shadow Control register 2 and can be controlled at 16K resolution. ROM chip-select generation for the entire 64K range can be controlled via Shadow Control register 3. This address range is hardwired to be non-cacheable.
SHADOW (64K)	000F0000h 000FFFFFFh	This range has the same characteristics as that of 000C0000h-000C7FFFh segment, as described above. Shadow control for this address range is provided via Shadow Control register 3. If not shadowed in the main memory, cycles in this address range which are forwarded to the ISA bus will always results in an ROMCS# assertion. The cacheability of this address segment is controlled via Shadow Control register 3.
TOP OF ADDRESSABLE SDRAM MEMORY (1M)	00100000h	This address segment is mapped to the main memory with the exception of one hole that can optionally be opened in the range 1MBytes to 16 Mytes. The address range defined for the hole is mapped to the expansion buses and is described later in this section. The addressable SDRAM memory can be different from the populated memory due to the memory remapping and the frame buffer. This is described in more detail in a later section. With the exception of the memory holes, this address range has the same characteristics as the 0-640K (compatible DOS memory) range.
TOP OF ADDRESSABLE SDRAM MEMORY (4G-256K)	FFFC0000h	All cycles above the addressable SDRAM memory are forwarded to the expansion buses. Host access in this range initiates a PCI cycle and if unclaimed by a PCI slave, they are forwarded to ISA. Note that the ISA address space is only 16M. Higher addresses are aliased to this 16M space.
ROM ALIAS (4G-64K)	FFFF0000 FFFFFFFFh	This address segment is an alias of the 64K segment located at F0000h-FFFFFFh and has the same attributes except that this segment can never be shadowed into the SDRAM memory. This is also true for address E0000h, D0000h and C0000h provided I/O register Index 51h (see Section 9.5.2 .) is set correctly.

6.3.1. MEMORY HOLE

The ISA bus is accessed through the PCI bus using subtractive decode and regardless of the address size. The ISA address bus is 24 bits wide and any PCI bus cycles which is not asserted by a PCI device will be transferred to the ISA. The 8 high bits of the address are just discarded. For example, the 0xFFFFFFF0 address generated by the CPU at reset is going to be forwarded to ISA at address 0xFFFFF0.

Through this way, an ISA device can be accessed at multiple locations in the CPU address space. Any 0x????xxxx address is going to be forwarded at ISA address 0x00xxxxxx. Only the true original address is processed on PCI bus and the address should be chosen in order to not conflict with any PCI device.

If other devices are present on the ISA memory address space, an external decoding have to be implemented to avoid conflicts with this device.

6.3.2. SMM MEMORY

The STPC uses the physical memory behind the CPU address range A0000h - B0000h for the SMM memory. The SMM base address register inside CPU needs to be programmed to A0000h. The initialization of the SMM memory is controlled by RAM System management register and redirects the CPU A0000h-B0000h address range to SMM memory. After the initialization, SMM memory can only be accessed when SMI/ACT# is active. The cacheability of this segment is hardwired to 0.

6.3.3. ADDRESSABLE SDRAM MEMORY

Addressable SDRAM memory is a function of the size of populated SDRAM, the size of graphic memory, the size of memory hole, and the shadow control of D0000h-DFFFFh and E0000h-EFFFFh segments.

TOPM = The size of total physical SDRAM is defined by SDRAM Bank 3 Register.

TOGM = The size of graphic memory is defined by Graphic memory size register.

MHOLE_SIZE = The size of memory hole defined by Memory Hole Control register.

REMAP_SIZE = 128KB, if none of the 8 x 16KB-segments of D0000h-EFFFFh is enabled for shadow, or 0KB, if any of the 8 x 16KB-segments of D0000h-EFFFFh is enabled for shadow.

The addressable SDRAM memory =

TOPM - TOGM + MHOLE_SIZE + REMAP_SIZE

Note that this is an example and TOGM may not apply to all systems.

6.3.4. CPU ADDRESS TO SDRAM ADDRESS MAPPING

The CPU's concept of a physical address is a logical address to the STPC and is remapped to a SDRAM physical address. This section refers to the CPU's physical address as the "CPU address" and to the SDRAM's physical address as the "SDRAM address".

STPC also defines a memory hole to allow the existence of memory devices on the PCI or ISA buses. The size of the CPU address space is increased by these memory holes, if they exist. CPU address space D0000h to EFFFFh is mapped to the add-in card BIOS area. If this ROM space is not shadowed, then the CPU address space is increased by another 128 KBytes (also see [Section 6.6.1.](#)).

For example:

Total populated SDRAM = 4 MBytes

Frame buffer size = 256 KBytes*

Memory hole size = 1 MByte

Memory hole starting address = 200000h



HOST INTERFACE

Shadow feature for D0000h to EFFFFh = disabled

The total CPU memory = 4 MBytes - 256 KBytes + 1 MByte + 128 KBytes = 4 MBytes plus 896 KBytes

Since the frame buffer is 256 KBytes*, the system memory is reduced by 256 KBytes and becomes 3 MBytes plus 768 KBytes. Since a 1 MByte memory hole exists, the CPU address space is increased by 1 MByte and becomes 4 MBytes plus 768 KBytes. The CPU address between 3 MBytes plus 768 KBytes and 128 MBytes above this is mapped to the memory hole.

Since the shadowing of the CPU address range D0000h to EFFFFh reserved for add-on card BIOS is not enabled, the CPU memory is increased by 128 KBytes to make use of this SDRAM space that no device accesses. The total CPU memory then becomes 4 MBytes plus 896 KBytes.

* Not applicable to STPC Elite or Vega.

6.4. IO ADDRESS MAP

Table 6-1. IO Map Space

IO address	Description	Notes
0000h-000Fh	8237 DMA controller 1 registers.	1
0020h-0021h	8259 Interrupt controller 1 registers.	
0022h	STPC specific configuration registers index port	
0023h	STPC specific configuration registers data port	
0040h-0043h	8254 Timer/Counter registers.	1
0060h-0064h	Keyboard shadow registers.	1
0070h-0071h	NMI Mask control registers.	1
0080h-008Fh	DMA Page registers.	
00A0h-00A1h	8259 Interrupt controller 2 registers.	1
061h	ISA standard Port B.	1
00C0h-00DFh	8237 DMA controller 2 registers.	1
0CF8h	PCI configuration Address register.	
0CFCh-0CFFh	PCI configuration Data register.	

The STPC implements a number of registers in IO address space. This is visible in the registers with access = 0022h/0023h. These registers use the index and data programming system where the index to which the data is to be written to is programmed in register 0022h and the data is written to register 0023h.

These registers occupy the map in the IO space in the table above:[Table 6-1](#)

Notes:

1. This address range is partially decoded. Refer to the Register Description section for more details.
2. This address is occupied only if the STPC is strapped to look like a mother-board VGA.

6.4.1. PCI CONFIGURATION ADDRESS MAP:

The STPC occupies Device number 0 slot on the PCI bus and implements a number of registers in PCI configuration address space. These registers occupy the following map (see [Table 6-2](#)) :

Table 6-2. PCI Configuration Address Space

Offset	Description
00h-01h	Vendor Identification register
02h-03h	Device Identification register
04h-05h	PCI Command register
06h-07h	PCI Status register
08h	PCI Revision ID register
40h	PCI Control register

6.5. CACHE RELATED REGISTERS

The STPC supports two caching modes, write-through and write-back. For both modes, sdram read accesses are copied into the cache, and future accesses then return the cache copy with no access to sdram. The situation is different for write access; for the write-through mode, a write updates both the sdram and the cache, whereas in the the write-back mode, a write updates the cache only with the sdram update taking place later. The write-back mode offers improved performance over the write-through mode.

Two cache levels are generally available, level 1 (L1), within the CPU core, and level 2 (L2), within the chipset core (between the CPU and the sdram Controller). For the STPC product range, the L2 cache controller is included but as the required external connection pins are not provided, it is not usable. The L2 cache control registers described below are however used to configure the CPU L1 Cache architecture.

6.5.1. CACHE ARCHITECTURE REGISTER 0

This register controls various attributes of the L2 and L1 cache.

Cash_Arc0			Access =			Regoffset = 0x20h	
7	6	5	4	3	2	1	0
Rsv	BAO	L1 WB	SRAM		L2 B	L2 WBC	L2 BC
Default value after reset = 00000000h							

Bit Number	Mnemonic	Description
Bit 7	Rsv	Reserved
Bit 6	BAO	Burst addressing order. Should be set to 0.
Bit 5	L1 WB	L1 write back indication. 0: Not supported 1: Supported
Bit 4-3	SRAM	SRAM type. These bits control the type of SRAMs used to construct L2 cache. (See Table 6-3)
Bit 2	L2 B	Number of L2 banks. When programmed to 2 banks, L2 interleaving is enabled. 0: One bank 1: Two banks
Bit 1	L2 WBC	L2 write back control. 0: Write through 1: Write back
Bit 0	L2 BC	L2 cache enable. 0: Disabled 1: Enabled



Table 6-3. Bits 4-3 SRAM Type

Bit 4	Bit 3	L2 cache SRAM type
0	0	asynchronous SRAM
0	1	synchronous burst SRAM
1	0	synchronous burst pipelined SRAM
1	1	reserved



6.5.2. CACHE ARCHITECTURE REGISTER 1

This register controls various attributes of L2 cache.

Cash_Arc1			Access =			Regoffset = 0x21h	
7	6	5	4	3	2	1	0
L2 CS			Rsv	Rsv		R AWE	Rsv
Default value after reset = 00000000h							

Bit Number	Mnemonic	Description
Bits 7-5	L2 CS	L2 cache size. (See Table 6-4)
Bit 4	Rsv	Reserved
Bits 3-2	Rsv	Reserved
Bit 1	R AWE	Read around write enable. 0: Reads can not proceed around any posted writes 1: Reads can go around a posted write if it is to a different address to the posted writes
Bit 0	Rsv	Reserved.

Table 6-4. L2 Cache Size

Bit 7	Bit 6	Bit 5	L2 Cache Size
0	0	0	64Kb
0	0	1	128Kb
0	1	0	256Kb
0	1	1	512Kb
1	0	0	1 MB
1	0	1	2 MB



6.5.3. CACHE ARCHITECTURE REGISTER 2

Cash_Arc2			Access =			Regoffset = 0x22h	
7	6	5	4	3	2	1	0
Rsv	SHDD	CWEPW	CDHAWE	BAWS		TAWS	
Default value after reset = 11111111h							

Bit Number	Mnemonic	Description
Bit 7	Rsv	Reserved.
Bit 6	SHDD	Slow host data driver. 0: Fast, One clock to drive the HD bus 1: Slow, two clocks to drive HD bus
Bit 5	CWEPW	Cache write enable pulse width. 0: 1 clock wide 1: 1.5 clocks wide Applicable to asynchronous SRAMs only. Must be '0' for synchronous SRAMs.
Bit 4	CDHAWE	Cache data hold after write enable. 0: Data removed in the same clock as write enable trailing edge 1: Data is kept valid for 1 extra clock after write enable Must be a '1' if 1.5 clocks wide write enable pulse width is selected via bit 5 above.
Bits 3-2	BAWS	Burst access wait states. (See Table 6-5)
Bits 1-0	TAWS	Tag access wait states. (See Table 6-6)

Table 6-5. Burst Access Wait States

Bit 3	Bit 2	Burst access wait states
0	0	fastest
0	1	1 clock slower than fastest
1	0	2 clocks slower than fastest
1	1	3 clocks slower than fastest

Table 6-6. Tag Access Wait States

Bit 1	Bit 0	Tag access wait states
0	0	fastest
0	1	1 clock slower than fastest
1	0	2 clocks slower than fastest
1	1	3 clocks slower than fastest



6.6. ADDRESS DECODE RELATED REGISTERS

The following registers are all 8-bit. They are accessed by setting the Configuration Index Port (22h) to the Configuration Index (C.I.) shown, and then reading or writing the appropriate values from the Configuration Register Data Port (23h).

6.6.1. MEMORY HOLE CONTROL REGISTER

This 8-bit register defines the enable, size, and starting address of memory hole. Any memory accesses to this memory hole are directed to PCI/ISA bus.

MEM_HOLE				Access =		Regoffset = 0x24h	
7	6	5	4	3	2	1	0
MHE	MHS			MHSA			
Default value after reset = 00000000h							

Bit Number	Mnemonic	Description
Bit 7	MHE	Memory Hole Enable. This bit controls the enable of memory hole function 0 = disabled 1 = enabled
Bits 6-4	MHS	Memory Hole Size. These bits control the size of memory hole (See Table 6-7)
Bits 3-0	MHSA	Memory Hole Start Address. These bits control the bits 23-20 of the memory hole starting address. The memory hole starting address must be aligned to the hole size.

Table 6-7. Memory Hole Size

Bit 6	Bit 5	Bit 4	Memory Hole Size
0	0	0	1 MB
0	0	1	2 MB
0	1	1	4 MB
1	1	1	8 MB
others			reserved

Programming notes:

This memory hole is also non-cacheable.



6.6.2. SHADOW CONTROL REGISTER 0

This 8-bit register controls the read/write attributes of the memory located at C0000h-CFFFFh. Each 16k of the whole 64k is controlled by 2 bits, one for read and one for write.

SHADOW_0				Access =		Regoffset = 0x25h	
7	6	5	4	3	2	1	0
RC1	WC1	RC2	WC2	RC3	WC3	RC4	WC4
Default value after reset = 0000000h							

Bit Number	Mnemonic	Description
Bit 7	RC1	Read Control CC000h-CFFFFh. This bit controls the read attribute of the CC000h-CFFFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 6	WC1	Write Control CC000h-CFFFFh. This bit controls the write attribute of the CC000h-CFFFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle
Bit 5	RC2	Read Control C8000h-CBFFFh. This bit controls the read attribute of the C8000h-CBFFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 4	WC2	Write Control C8000h-CBFFFh. This bit controls the write attribute of the C8000h-CBFFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle
Bit 3	RC3	Read Control C4000h-C7FFFh. This bit controls the read attribute of the C4000h-C7FFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 2	WC3	Write Control C4000h-C7FFFh. This bit controls the write attribute of the C4000h-C7FFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle

Bit Number	Mnemonic	Description
Bit 1	RC4	Read Control C0000h-C3FFFh. This bit controls the read attribute of the C0000h-C3FFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 0	WC4	Write Control C0000h-C3FFFh. This bit controls the write attribute of the C0000h-C3FFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle

Programming Notes:

There is single cacheability bit for the 32k Video BIOS segment (C0000h-C7FFFh) located in Shadow Control register 2. C7FFFh-CFFFFh segment has the cacheability bit hardwired to '1' (enabled). If shadow is enabled for read/write cycles, read from and write to this area are directed to the system memory. Or else the cycles are forwarded to the expansion buses.

6.6.3. SHADOW CONTROL REGISTER 1

Similar to Shadow Control Register 0, this 8-bit register controls the read/write attributes of the memory located at D0000h-DFFFFh.

SHADOW_1				Access =		Regoffset = 0x26h	
7	6	5	4	3	2	1	0
SRC	SWC	SWC	SWC	SRC	SWC	SRC	SWC
Default value after reset = 00000000h							

Bit Number	Mnemonic	Description
Bit 7	SRC	Shadow Read Control DC000h-DFFFFh. This bit controls the read attribute of the DC000h-DFFFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 6	SWC	Shadow Write Control DC000h-DFFFFh. This bit controls the write attribute of the DC000h-DFFFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle
Bit 5	SWC	Shadow Write Control D8000h-DBFFFh. This bit controls the read attribute of the D8000h-DBFFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 4	SWC	Shadow Write Control D8000h-DBFFFh. This bit controls the write attribute of the D8000h-DBFFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle
Bit 3	SRC	Shadow Read Control D4000h-D7FFFh. This bit controls the read attribute of the D4000h-D7FFFh memory 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 2	SWC	Shadow Write Control D4000h-D7FFFh. This bit controls the write attribute of the D4000h-D7FFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle
Bit 1	SRC	Shadow Read Control D0000h-D3FFFh. This bit controls the read attribute of the D0000h-D3FFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 0	SWC	Shadow Write Control D0000h-DFFFFh. This bit controls the write attribute of the D0000h-DFFFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle

Programming Notes: This entire 64K segment has the cacheability bit hardwired to '0' (disabled)



6.6.4. SHADOW CONTROL REGISTER 2

Similar to Shadow Control Register 0, this 8-bit register controls the read/write attributes of the memory located at E0000h-EFFFFh.

SHADOW_2

Access =

Regoffset = 0x27h

7	6	5	4	3	2	1	0
RC	WC	RC	WC	RC	WC	RC	WC
Default value after reset = 00000000h							

Bit Number	Mnemonic	Description
Bit 7	RC	Read Control EC000h-EFFFFh. This bit controls the read attribute of the EC000h-EFFFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 6	WC	Write Control EC000h-EFFFFh. This bit controls the write attribute of the EC000h-EFFFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle
Bit 5	RC	Read Control E8000h-EBFFFh. This bit controls the read attribute of the E8000h-EBFFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 4	WC	Write Control E8000h-EBFFFh. This bit controls the write attribute of the E8000h-EBFFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle
Bit 3	RC	Read Control E4000h-E7FFFh. This bit controls the read attribute of the E4000h-E7FFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 2	WC	Write Control E4000h-E7FFFh. This bit controls the write attribute of the E4000h-E7FFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle
Bit 1	RC	Read Control E0000h-E3FFFh. This bit controls the read attribute of the E0000h-E3FFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 0	WC	Write Control E0000h-EFFFFh. This bit controls the write attribute of the E0000h-EFFFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle

Programming Notes: This entire 64K segment has the cacheability bit hardwired to '0' (disabled)

6.6.5. SHADOW CONTROL REGISTER 3

This 8-bit register controls the cacheability attributes of C0000h-C7FFFh and F0000h-FFFFFFh shadow segments.

SHADOW_3

Access =

Regoffset = 0x28h

7	6	5	4	3	2	1	0
SMRAM	CCF	CCC	Rsv	RsvMHR	256MB	RCF	WCF

Default value after reset = 00000000h

Bit Number	Mnemonic	Description
Bit 7	SMRAM	SMRAM Initialization Enable. This bit controls whether CPU accesses in A0000h-BFFFFh address range are decoded as VGA frame buffer access or SMRAM access. 0 = A0000h-BFFFFh is interpreted as VGA frame buffer access 1 = A0000h-BFFFFh is interpreted as SMRAM access. The STPC allows for 128KBytes of SMRAM. Physically this memory is located in the system memory behind the higher address range. This area of the system memory is normally unused since this address range is normally mapped to frame buffer which has its own memory. When the CPU is operating in SMM, accesses in the range of A0000-BFFFFh goes to SMRAM instead of VGA frame buffer. The rest of the address map remains unchanged. The address range A0000h-BFFFFh is always non-cacheable.
Bit 6	CCF	Cache Control F0000h-FFFFFFh. This bit controls the cacheability of F0000h-FFFFFFh block when the shadow function is enabled. 0 = cacheability disabled 1 = cacheability enabled
Bit 5	CCC	Cache Control C0000h-C7FFFh. This bit controls the cacheability of C0000h-C7FFFh block when the shadow function is enabled. 0 = cacheability disabled 1 = cacheability enabled
Bits 4	Rsv	Reserved.
Bit 3	MHR	HCLK - MCLK Relation, Should be set to 0 if MCLK is greater or equal to HCLK. Should be set to 1 if HCLK is greater than MCLK
Bit 2	256MB	Enable 256MByte support
Bit 1	RCF	Read Control F0000h-FFFFFFh. This bit controls the read attribute of F0000h-FFFFFFh memory. 0 = shadow disabled for read cycle 1 = shadow enabled for read cycle
Bit 0	WCF	Write Control F0000h-FFFFFFh. This bit controls the write attribute of F0000h-FFFFFFh memory. 0 = shadow disabled for write cycle 1 = shadow enabled for write cycle



Programming notes:

The rest of the shadow RAM segments have the cacheability bits hardwired to '0' (disabled). This register also provides control over the address range for which ROM chip-select (ROMCS#) will be asserted allowing various BIOSes (system, video, disk etc.) to be implemented in a single part. Bit 7 of this register also provides accessibility to the SMM mode RAM (SMRAM).

6.7. HOST SDRAM CONTROLLER REGISTERS

The STPC manages 4 Memory Banks (if DIMM sockets are used they can be populated with either single or double sided 64-bit data DIMMs). For SDRAM densities are supported see the datasheet [Section 6.3.3](#).

Configuration registers 30-33 provide the top addresses for each bank. Any bank can be skipped by the top addresses of two consecutive banks having the same address.

6.7.1. MEMORY BANK 0 REGISTER - C.I. 30H (MEMORY__BANK0)

This 8-bit register controls the top address of memory bank 0. Register bit 7-0 corresponding to memory address bits 27-20.

Bank 0 Top Address = Memory Bank0 size in MBytes -1.

Bank 1 Top Address = Memory Bank0 + Memory Bank1 size in MBytes -1

This register defaults to 07h.

Example 1:

Memory Bank0 = 4MB

Memory Bank1 = 4MB

Bank 0 Top Address = 4 -1 = 3 = 03h

Bank 1 Top Address = 4 + 4 - 1 = 07h

Bank 2, 3 Top Address = 07h

Example 2: for use with double sided DIMMs

Memory Bank0 = 32MBytes (dbl. sided DIMMS)

Memory Bank1 = 32MBytes (dbl. sided DIMMS)

Bank 0 Top Address = 16 - 1 = 15 = 0Fh

Bank 1 Top Address = 16 + 16 - 1 = 31 = 1Fh

Bank 2 Top Address = 32 + 16 - 1 = 47 = 2Fh

Bank 3 Top address = 48 + 16 - 1 = 63 = 3Fh

6.7.2. MEORY BANK 1 REGISTER - C.I. 31H (MEMORY__BANK1)

This register controls the top address of memory bank 1.

6.7.3. MEMORY BANK 2 REGISTER - C.I. 32H (MEMORY__BANK2)

This register controls the top address of memory bank 2.

6.7.4. MEMORY BANK 3 REGISTER - C.I. 33H (MEMORY__BANK3)

This register controls the top address of memory bank 3.

6.7.5. SDRAM REFRESH REGISTER

This refresh register also contains a number of host clock settings for the SDRAM refresh interval.

SDRAM_Ref			Access = 0022h/0023h			Regoffset = 039h	
7	6	5	4	3	2	1	0
RE	RC						
Default value after reset = 30h							

Bit Number	Mnemonic	Description
Bit 7	RE	Refresh Enable. This bit must be programmed to '0' for normal operation
Bits 6-0	RC	Refresh Cycle. (HCLK frequency in MHz * 15.6us) >> 4

* Examples: (rounded down to nearest integer)

round_down((75MHz * 15.6us) >> 4) = 73 = 49h

round_down((66MHz * 15.6us) >> 4) = 65 = 41h

round_down((60MHz * 15.6us) >> 4) = 58 = 3Ah

round_down((50MHz * 15.6us) >> 4) = 48 = 30h

Programming notes:

The refresh interval should be reset to the smallest likely run time value (typically 48 HCLKs) to provide warm up cycles for the SDRAM.

A refresh request is generated whenever this register is written to without setting the refresh enable bit.



6.8. ACCESSING CONFIGURATION REGISTERS

The Host interface and the Local Bus Interface are programmed identically. To access all the internal configuration registers, the programmer will need to program the Index (address) and data registers of the required interface through port 22h/23h. The principle of the programming is to fix the address of device to a location that the user requires and to always address it at that location. The steps required to access any of the internal registers are as follows:

1. Select the interface you want to programme. Each interface is described as a device and both have a number. The Local Bus device number is 6 and the Host device number is 7. The below example code describes how the devices are accessed and the Host device is used.

2. Select Host interface base programming option in RBI, by writing 0x00 and 0x07 (0x06 for the Local Bus, see [Section 11.4.](#) for more details) in register 0x11 (index value 0x11 accessed through I/O 22h/23h address) and 0x10 respectively.

```
IOWRITE8(0x22,0x10);
```

```
IOWRITE8(0x23,0x07);
```

```
IOWRITE8(0x22,0x11);
```

```
IOWRITE8(0x23,0x00);
```

3. Select the Host interface address. Assume that HOST_BASE is the address of the Host interface I/O space.

```
IOWRITE8(0x22,0x12);
```

```
IOWRITE8(0x23,(HOST_BASE &0xFF) | 0x03);
```

```
IOWRITE8(0x22,0x13);
```

```
IOWRITE8(0x23,HOST_BASE >>8);
```

The host interface registers are then accessed with HOST_BASE as the index register and HOST_BASE+4 as the data register, as shown below.

4. Writing into any Internal Register of the Host Interface:

```
IOWRITE8(HOST_BASE,offset);
```

```
IOWRITE32(HOST_BASE+4,data);
```

Here the "offset" index address is as mentioned in the register table shown above. The 32-bit "data" is written into the register.

5. Reading from any internal register of Host Input

```
IOWRITE8(HOST_BASE,offset);
```

```
IOREAD32(HOST_BASE+4,data);
```

Here the "offset" index address is as mentioned in the register tables shown above. The 32-bit "data" is expected to be read from the internal register.

One constraint is that the Local Bus address must be set at multiples of 8h.

6.1 GPCLK Control registers

These registers control the General Purpose Clock which can be used to clock any external device of the STPC.

6.1.1. GPCLK Control Register 00

This is one of the 4-pairs of General Purpose Clock Control registers. This pair (00 and 01) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 00.

GPCLK00

Access = 022h/023h

Regoffset =42h

7	6	5	4	3	2	1	0
Rsv	4BD				8BN		
Default value after reset = 0x76h							

Bit Number	Mnemonic	Description
Bit 7	Rsv	Reserved.
Bits 6-3	4BD	This is the 4-bit M (divisor) value of the General Purpose Clock Synthesiser.
Bits 2-0	8BN	These are bits 7-5 of the 8 bit N (multiplier) of the General Purpose Clock synthesiser.

Programming notes:

This register defaults to 0x76h at reset. This value when combined with the default value of the other half of this pair results in a General Purpose Clock of 25.18 MHz assuming 14.318 MHz oscillator clock as the reference input.

HOST INTERFACE

6.1.2. GPCLK control register 01

This is one of the 4-pairs of General Purpose Clock Control registers. This pair (00 and 01) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 00.

GPCLK01				Access = 022h/023h			Regoffset =43h	
7	6	5	4	3	2	1	0	
3BP0		8BN				3BP1		
Default value after reset = 0x95h								

Bit Number	Mnemonic	Description
Bit 7	3BP0	This is the bit 0 of the 3-bit P (exponent) value of the General Purpose Clock synthesiser.
Bits 6-2	8BN	These are bits 4-0 of the 8-bit N (multiplier) value of the General Purpose Clock Synthesiser.
Bits 1-0	3BP1	These are bits 2-1 of the 3-bit P (exponent) value of the General Purpose Clock synthesiser.

Programming notes:

This register defaults to 0x95h at reset. This value when combined with the default value of the other half of this pair results in a General Purpose Clock of 25.18 MHz assuming 14.318 MHz oscillator clock as the reference input.



6.1.3. GPCLK control register 10

This is one of the four pairs of General Purpose Clock control registers. This pair (10 and 11) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 01.

GPCLK10				Access = 022h/023h				Regoffset =44h				
7	6	5	4	3	2	1	0					
Rsv	4BM				8BN							
Default value after reset = 0x76h												

Bit Number	Mnemonic	Description
Bit 7	Rsv	Reserved.
Bits 6-3	4BM	This is the 4-bit M (divisor) value of the General Purpose Clock Synthesiser.
Bits 2-0	8BN	These are bits 7-5 of the 8 bit N (multiplier) of the General Purpose Clock Synthesiser.

Programming notes:

This register defaults to 0x76h at reset. This value when combined with the default value of the other half of this pair results in a General Purpose Clock of 28.0 MHz assuming 14.318 MHz oscillator clock as the reference input.



HOST INTERFACE

6.1.4. GPCLK control register 11

This is one of the four pairs of General Purpose Clock Control registers. This pair (10 and 11) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 01.

GPCLK11				Access = 022h/023h			Regoffset =45h	
7	6	5	4	3	2	1	0	
3BP0		8BN				3BP1		
Default value after reset = 0xEDh								

Bit Number	Mnemonic	Description
Bit 7	3BP0	This is the bit 0 of the 3-bit P (exponent) value of the General Purpose Clock Synthesiser.
Bits 6-2	8BN	These are bits 4-0 of the 8-bit N (multiplier) value of the General Purpose Clock Synthesiser.
Bits 1-0	3BP1	These are bits 2-1 of the 3-bit P (exponent) value of the General Purpose Clock Synthesiser.

Programming notes:

This register defaults to 0xEDh at reset. This value when combined with the default value of the other half of this pair results in a General Purpose Clock of 28.0 MHz assuming 14.318 MHz oscillator clock as the reference input.



6.1.5. GPCLK control register 20

This is one of the four pairs of General Purpose Clock Control registers. This pair (20 and 21) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 10.

GPCLK20				Access = 022h/023h				Regoffset =46h				
7	6	5	4	3	2	1	0					
Rsv	4BM				8BN							
Default value after reset = 0x5Bh												

Bit number	Mnemonic	Description
Bit 7	Rsv	Reserved.
Bits 6-3	4BM	This is the 4-bit M (divisor) value of the General Purpose Clock Synthesiser.
Bits 2-0	8BN	These are bits 7-5 of the 8 bit N (multiplier) of the General Purpose Clock Synthesiser.

Programming notes

This register defaults to 0x5Bh at reset. This value when combined with the default value of the other half of this pair results in a General Purpose Clock of 40 MHz assuming 14.318 MHz oscillator clock as the reference input.



HOST INTERFACE

6.1.6. GPCLK control register 21

This is one of the four pairs of General Purpose Clock Control registers. This pair (20 and 21) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 10.

GPCLK21			Access = 022h/023h			Regoffset =47h	
7	6	5	4	3	2	1	0
3BP0	8BN					3BP1	
Default value after reset = 0x6Dh							

Bit Number	Mnemonic	Description
Bit 7	3BP0	This is the bit 0 of the 3-bit P (exponent) value of the General Purpose Clock Synthesiser.
Bits 6-2	8BN	These are bits 4-0 of the 8-bit N (multiplier) value of the General Purpose Clock Synthesiser.
Bits 1-0	3BP1	These are bits 2-1 of the 3-bit P (exponent) value of the General Purpose Clock Synthesiser.

Programming notes:

This register defaults to 0x6Dh at reset. This value when combined with the default value of the other half of this pair results in a General Purpose Clock of 40 MHz assuming 14.318 MHz oscillator clock as the reference input.



6.1.7. GPCLK control register 30

This is one of the four pairs of General Purpose Clock control registers. This pair (30 and 31) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 11.

GPCLK30				Access = 022h/023h			Regoffset =48h	
7	6	5	4	3	2	1	0	
Rsv	4BM				8BN			
Default value after reset = 0x6Eh								

Bit Number	Mnemonic	Description
Bit 7	Rsv	Reserved.
Bits 6-3	4BM	This is the 4-bit M (divisor) value of the General Purpose Clock Synthesiser.
Bits 2-0	8BN	These are bits 7-5 of the 8 bit N (multiplier) of the General Purpose Clock Synthesiser.

Programming notes:

This register defaults to 0x6Eh at reset. This value when combined with the default value of the other half of this pair results in a General Purpose Clock of 60 MHz assuming 14.318 MHz oscillator clock as the reference input.



6.1.8. GPCLK control register 31

This is one of the four pairs of General Purpose Clock control registers. This pair (30 and 31) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 11.

GPCLK31			Access = 022h/023h			Regoffset =49h	
7	6	5	4	3	2	1	0
3BP0	8BN					3BP1	
Default value after reset = 0x69h							

Bit Number	Mnemonic	Description
Bit 7	3BP0	This is the bit 0 of the 3-bit P (exponent) value of the General Purpose Clock Synthesiser.
Bits 6-2	8BN	These are bits 4-0 of the 8-bit N (multiplier) value of the General Purpose Clock Synthesiser.
Bits 1-0	3BP1	These are bits 2-1 of the 3-bit P (exponent) value of the General Purpose Clock Synthesiser.

Programming notes:

This register defaults to 0x69h at reset. This value when combined with the default value of the other half of this pair results in a General Purpose Clock of 60 MHz assuming 14.318 MHz oscillator clock as the reference input.



7. SDRAM CONTROLLER

7.1. INTRODUCTION

This chapter describes the mapping of the CPU memory and IO address spaces.

Details of memory shadowing and cachability by software control and the Memory Hole for ISA BIOS are given. The actual interface to the external SDRAM modules is presented. Also introduced in this chapter are the PCI configuration space mapping registers, further details are in the chapter relating to the PCI Bus Controller.

7.2. MEMORY CONTROLLER

The STPC handles the memory data bus directly, controlling from 8 MBytes to 256 MBytes. The SDRAM controller supports accesses to the Memory Banks to/from the CPU (via the host) and the local Bus which can be populated with either single-sided or double-sided 64-bit memory devices. Parity is not supported.

The SDRAM controller only supports 64 bit wide Memory Banks.

The SDRAM Controller supports buffered or unbuffered SDRAM but not EDO or FPM modes. SDRAMs must support Full Page Mode Type access.

The STPC Memory Controller provides various programmable SDRAM parameters to allow the SDRAM interface to be optimised for different processor bus speeds SDRAM speed grades and CAS Latency.

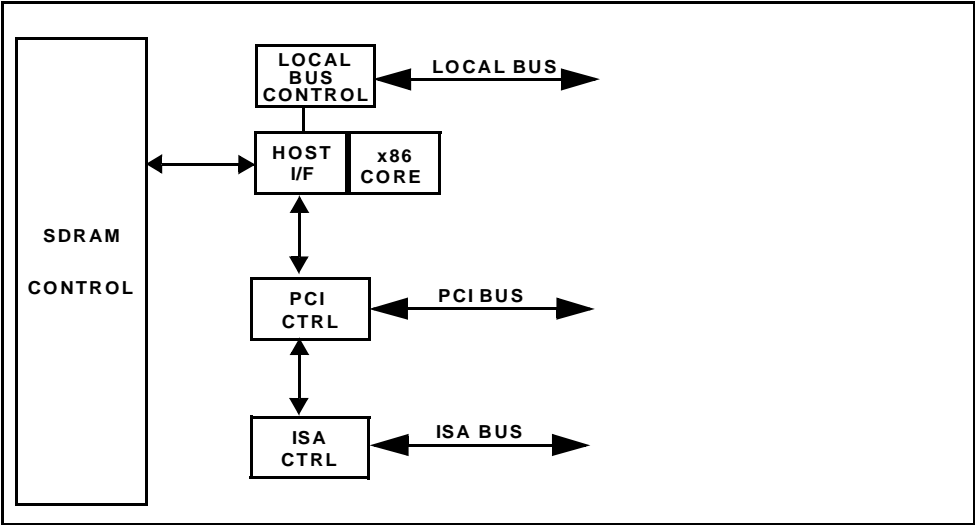


Figure 7-1. Memory Controller Interface Block Diagram

7.3. SDRAM REGISTER ACCESS

These registers are used to configure the SDRAM controller.

7.3.1. REGISTER 0

This is a 31-bit configuration register for the SDRAM controller block:

MEM_REG0				Access =								Regoffset = 84C6000h			
	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RCASLCY			REGD	MRS R	RASo	LHDI	Rsv	Rsv	CASLat			Config		PRA
Default value after reset = 31x32198376h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRA	Rsv				RRW			BT	BL			RCT			
Default value after reset = 31x32198376h															

Bit Number	Mnemonic	Description
Bits 30-28	RCASLCY	Read CAS Latency (RCASLCY) should be equal to CASLCY but is left programmable for debug purpose.
Bit 27	REGD	Registered DIMM , Indicate if we use registered DIMMs '1' or not '0'.
Bit 26	MRSR	Mode Register Set Request , If set to 1, we update the SDRAM chips corresponds to the value programmed Bits [16:0].
Bit 25	RASo	RAS on/off , When we finish a read or a write, if set to 1 go into RACTIVE, if set to 0, go into PRECHARGE and the IDLE.
Bit 24	LHDI	Latch_Host_Data_In for host input data. If set to 0, select MD[63:0] directly from SDRAM, if set to 1, select latched MD[63:0] by RDCLK (see bits 3-0 in Register 1).
Bit 23	Rsv	Reserved.
Bit 22	Rsv	Reserved
Bits 21-19	CASLat	CAS Latency This is abbreviated as CL in SDRAM datasheets and is defined as the number of clock cycles for the data held for after CAS goes low.
Bits 18-17	Config	CONFIGURATION of a DIMM. (See Table 7-1) These bits are used to determine the maximum burst length (full page burst). If the DIMMS are populated with a different kind of memory, the 2 bits are programmed to maximise the burst length using the minimum amount of the memory.
Bits 16-15	PRA	Precharge to Row Active number of cycles (called tRP in datasheets).



Bit Number	Mnemonic	Description
Bits 14-11	Rsv	Reserved. Should be set to '0000'.
Bits 10-8	RRW	RACTIVE to Read/Write (Called tRCD in datasheets).
Bit 7	BT	Burst Type. Should be set to '0'.
Bits 6-4	BL	Burst Length. Should be set to '111'
Bits 3-0	RCT	Refresh Cycle Timing (Called tRC in SDRAM datasheets).

Table 7-1. Memory Bank Configuration

Bit 18	Bit 17	Memory Bank Configuration	Maximum Burst Length
0	0	[4Mx4]x16	1024
0	1	[2Mx8]x8	512
1	0	[1Mx16]x4	256
1	1	Reserved	



7.3.2. REGISTER 1

This 7-bit register is used for the read clock scheme. See Chapter 6.3 “Clock considerations” for more details. This delay can be set up to 3.5 ns beyond the 15ns required from the previous MCLKI edge.

MEM_REG1			Access =			Regoffset = 84C6004h	
	6	5	4	3	2	1	0
	BLS	CSMEM	MEM	RCDP			
Default value after reset = 000000h							

Bit Number	Mnemonic	Description
Bit 6	BLS	Bit Length Select: This selects whether the burst lgth of 1,2,4 or 8, or only full page. 1 = Burst mode 0 = Full page mode
Bit 5	CSMEM	CS_MEM16_OE It is programmed as bit 4 in case of 16Mbits only and is programmed to 1 otherwise.
Bit 4	MEM	MEM16_OE_ , This bit is set to '1' to get 16mA output enabled, set to '0' to get 8mA output enabled
Bits 3-0	RCDP	Read Clock Delay Programmation , 0000 for smallest delay to 1111 for largest delay



7.3.3. REGISTER 2 (MEM_REG2) 84C6008H

This 2-bit register is used to determine the type of SDRAM in use.

MEM_REG2	Access =	Regoffset = 84C6008h	
Empty		1	0
Empty		SDRAM	
Default value after reset = 000000h			

Bit Number	Mnemonic	Description
Bits 1-0	SDRAM	SDRAM type. (See Table 7-2)

Table 7-2. SDRAM Type

Bit 1	Bit 0	Description
0	0	16 Mbits SDRAMs
0	1	64 Mbits or 128 Mbits 2 bank SDRAM
1	0	64 Mbits or 128 Mbits 4 bank SDRAM
1	1	256Mbits 2 bank SDRAM



7.3.4. DDC CONTROL REGISTER DDCR

This register is used for Presence detect.

DDCR		Access = 022h/023h				Regoffset = 0x97h	
7	6	5	4	3	2	1	0
DDCWD		DDCRD		Rsv			Rsv
Default value after reset = 11111111h							

Bit Number	Mnemonic	Description
Bits 7-6	DDCWD	DDC Write Data. These two bits drive the DDC[1:0] open collector outputs. Writes to these bits affect the DDC[1:0] pins. The DDC[1:0] pins are open collector outputs which are externally pulled up. Thus, programming either of these bits to a one disables the output driver and allows the pin to act as an input whose status can be read via bits 5-4 of this register. Note that reads from these bits return the value of data last written to this register. This may not be the same as the data actually on the bus if another master is driving it. Bits 5-4 of this register accurately reflect the data on the bus no matter who is driving it.
Bits 5-4	DDCRD	DDC Read Data. These read-only bits return the read status of the DDC[1:0] pins.
Bits 3-1	Rsv	Reserved. These bits read are both readable and writable and must be programmed to ones to ensure future compatibility.
Bits 0	Rsv	Reserved. This bit must be programmed to '1' for correct operation.



7.4. MEMORY CLOCK REGISTERS

The MCLK register is used for Memory Clock control operations.

7.4.1. MCLK CONTROL REGISTER 0

MCLK00				Access =		Regoffset = 0x40h	
7	6	5	4	3	2	1	0
Uns	4-bitDIV				3-bitP		
Default value after reset = 0x5B							

Bit Number	Mnemonic	Description
Bit 7	Uns	Unused.
Bits 6- 3	4-bitDIV	This is the 4-bit M (divisor) value of the Memory synthesiser.
Bits 2-0	3-bitP	This is the 3-bit P (exponent) value of the Memory clock synthesizer.

This register defaults to 0x5B at reset. This value when combined with the default value of the other half of this pair results in a memory clock of 80.05 MHz assuming 14.318 MHz oscillator clock as the reference input.

For MCLK frequency programming values see table below:Table 7-3.

Note: If programming MCLK to a frequency not equal to HCLK, strap GPIO_Rx must be disabled (i.e. set to 0) to remove the synchronization between these two clock signals (see Vega Datasheet, STRAP OPTION section for further details).

$$MCLK = \frac{2 \times 14.31818 \times N}{M \times 2^P}$$

Constraints: $1 \leq M \leq 15$
 $1 \leq N \leq 255$
 $0 \leq P \leq 5$

$$1 \text{ MHz} \leq \frac{14.31818}{M} \leq 2 \text{ MHz}$$

$$200 \text{ MHz} \leq \frac{2 \times 14.31818 \times N}{M} \leq 622 \text{ MHz}$$



7.4.2. MCLK CONTROL REGISTER 1

MCLK01				Access =			Regoffset = 0x41h	
7	6	5	4	3	2	1	0	
8-bitN								
Default value after reset = 0xEC								

Bit Number	Mnemonic	Description
Bits 7- 0	8-bitN	These are bits 4-0 of the 8-bit N (multiplier) value of the Memory clock synthesiser.

This register defaults to 0xEC at reset. This value when combined with the default value of the other half of this pair results in a memory clock of 80.05 MHz assuming 14.318 MHz oscillator clock as the reference input.

For MCLK frequency programming values see [Table 7-3](#)

Table 7-3. MCLK Control Register Address 22 Index 40h, 41h

MHz	Reg1,Index 41h	Reg0, Index 40h	Actual Freq.	m	n	p
100	9Ah	5Ah	100.227260	11	154	2
95	92h	5Ah	95.020649	11	146	2
90	58h	3Ah	89.999989	7	88	2
85	5Fh	42h	85.014194	8	95	2
80	7Bh	5Ah	80.051643	11	123	2
75	88h	6Ah	74.895095	13	136	2
66	A6h	4Bh	66.022719	9	166	3
60	DAh	6Bh	60.026216	13	218	3
55	A9h	5Bh	54.994828	11	169	3
50	9Ah	5Bh	50.113630	11	154	3
45	58h	3Bh	44.999994	7	88	3
8	7Dh	75h	8.053978	7	125	5



8. PCI CONTROLLERS

8.1. INTRODUCTION

PCI Specification 2.1, from PCI-SIG, for further details of the PCI bus standard.

The PCI bus is the main data communication link to the STPC chip. Five PCI devices are present internally in the STPC, a host to PCI bridge (North Bridge), a PCI to ISA bridge (South Bridge), an IDE controller and a USB controller, LAN Controller. The STPC also contains a PCI arbiter which arbitrates between the bridges and for up to three external PCI devices. [Figure 8-1](#) shows the layout of the PCI controllers within the STPC. Please refer to *PCI Specification 2.1*, from PCI-SIG, for further details of the PCI bus standard.

The *North Bridge* translates the appropriate host bus I/O and Memory cycles to the PCI bus. It also supports generation of Configuration cycles on the PCI bus. The Configuration Address register allows for the remapping of host CPU I/O cycles, in the address range 0xCF8h-0xCFFh, to configuration cycles on the PCI bus.

The North Bridge, as a PCI bus agent (host bridge class), fully complies with PCI specification 2.1. The North Bridge also implements the PCI mandatory header registers in Type 0 PCI configuration space for easy porting of PCI-aware system BIOS. The North Bridge is assigned the Device Number 0xBh, which corresponds to the IDSEL on AD11 signal. PCI configuration registers of the North Bridge are accessible by the Type 0 PCI configuration cycles generated at Device number 0xBh.

The *South Bridge* controller responds to PCI configuration read and write transactions. The South Bridge is assigned the Device Number 0Ch, which corresponds to the IDSEL on AD12 signal. PCI configuration registers of the South Bridge are accessible by the Type 0 PCI configuration cycles generated by the North Bridge.

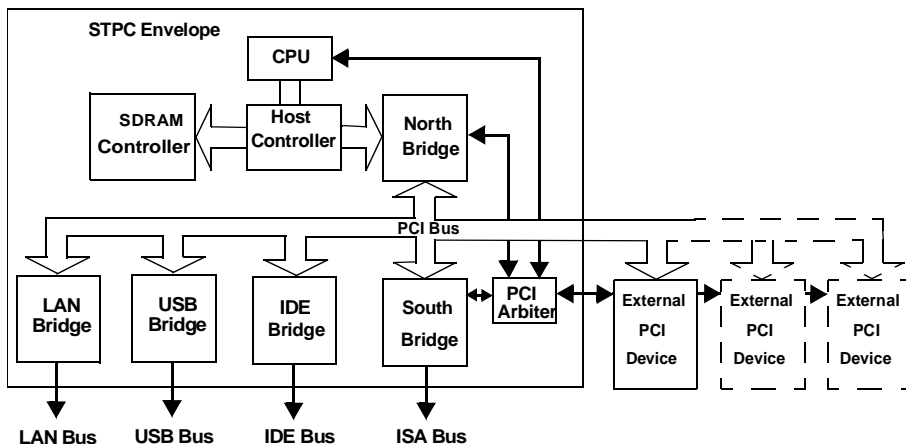


Figure 8-1. PCI Layout

8.1.1. PCI ADDRESS DECODE

ISA resources are accessed only via subtractive decode.

8.1.2. PCI ERROR HANDLING

Under control of South Bridge configuration registers, one or more of the following events can generate a 1 PCICLK long pulse on SERR#, which in turn can be made to generate an NMI to the CPU.

ISA initiated transaction ending in target abort.

8.1.3. PCI ARBITER

The PCI arbiter controls access to the PCI bus when several bus masters are present in the system. Whenever a further potential bus master needs to gain access to the bus, it asserts its request. The arbiter then asserts a system hold condition, which eventually causes a hold signal to be asserted to the CPU. The CPU finishes the current instruction, tristates the internal bus and asserts a hold acknowledge. This eventually causes the assertion of a system hold acknowledge. Once the system hold acknowledge is asserted, the arbiter asserts a grant to whichever requesting master is in the front of the line in a round-robin chain. When there are no requests pending or when the CPU is requesting the bus and it is in the front of the line, control of the bus is passed back to the CPU by the negation of the system hold condition.

8.2. ACCESSING THE PCI CONFIGURATION REGISTERS

The PCI configuration registers are accessed, from the CPU, using two 32-bit registers, mapped as I/O at CF8h and CFCh.

Each read from and write to the PCI configuration registers must be done by:

- Writing the 32-bit address of the PCI config. register using type 0 format at I/O CF8h.
- Reading or Writing 32-bit data at CFCh

All PCI configuration registers, inside the North and South bridges and all other external PCI devices, are seen from the CPU through those 2 x 32-bit registers.

An illustration of these registers is shown in [Table 8-1](#) & [Table 8-2](#).

Table 8-1. Register CF8h

31	30 ---- 24	23 ----- 16	15 ----- 11	10 ----- 8	7 ----- 2	1 0
Enable	Reserved	Bus number	Device number	Function number	Register number	0

Table 8-2. Register CFCh

31 ----- 24	23 ----- 16	15 ----- 8	7 ----- 0
Byte 3	Byte 2	Byte 1	Byte 0



8.3. PCI CONFIGURATION ADDRESS REGISTER

This is a 32-bit register accessible only via double-word IO read and write cycles.

Config_Address

Access = 0xCF8h

Regoffset =

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCI	Rsv							BN							
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DN					FN			RG						Rsv	
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bit 31	PCI	PCI configuration register access enable. When set to a '1', host CPU I/O cycles in address range CFCh-CFFh are converted to configuration cycles on the PCI bus. Otherwise if set to a '0', I/O cycles in this address range pass through as normal I/O cycles on the PCI bus.
Bits 30-24	Rsv	Reserved. Must be written to '0'. Read back as '0'.
Bits 23-16	BN	Bus Number. This field selects a specific bus number in the system. Bus Number 0 is assigned to the PCI bus directly behind the North Bridge. This field is driven on bits 23-16 of the AD bus during the address phase.
Bits 15-11	DN	Device Number. This field selects a specific device on the bus. During a Type-0 configuration cycle, this field is decoded to assert the appropriate IDSEL line as follows; The North Bridge Device Number 0xBh, which corresponds to IDSEL on AD11. The South Bridge Device Number 0xCh, which corresponds to IDSEL on AD12. LAN Bridge: Device Number 0xFh, which corresponds to IDSEL an AD15
Bits 10-8	FN	Function Number. During a PCI configuration cycle, this field is driven on bits 10-8 of the AD bus of the PCI during the address phase.
Bits 7-2	RG	Register Number. During a PCI configuration cycle, this field is driven on bits 7-2 of the AD bus during the address phase.
Bits 1-0	Rsv	Reserved. Must be written to a '0'. Reads back as '0'.

8.4. CONFIGURATION DATA REGISTER

This is a 32-bit register accessible via 8-bit, 16-bit and 32-bit IO read and write cycles.

Config_Data

Access = 0xCFCh

Regoffset =

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

8.5. NORTH BRIDGE CONFIGURATION REGISTERS

The STPC North Bridge configuration registers are accessed using the values below :

- Bus = 0h
- Device = 0Bh (IDSEL internally connected to PCI address line 11)
- Function = 0h (Host Bridge / PCI)

For example: Writing 80005800h at CF8h will access Vendor ID reg. index.

Table 8-3. North Bridge Reset Values

31 ----- 16 15 ----- 0

Device ID : 0201h		Vendor ID : 104Ah		00h
Status : 0280h		Command : 0007h		04h
Base class code: 06h	Sub class code: 00h	Program. Inter. Reg. : 00h	Revision : 00h	08h
	Header Type: 00h			0Ch
				...
Control Register : 00000000h				50h
Error Status Register : 00000000h				54h



8.5.1. NORTH BRIDGE PCI COMMAND REGISTER

This is the 16-bit PCI command register.

NB_Com																Access = 0xCF8h/0xCFCh				Regoffset = 0x4h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Rsv							S	AD	P	VGA	MW	ES	BS	ME	IO E								
Default value after reset = 0007h																							

Bit Number	Mnemonic	Description
Bits 15-9	Rsv	Reserved. These bits are hardwired to '0'. Writes have no effect on them.
Bit 8	S	SERR# enable. If this bit is set to a '1', the North Bridge may assert SERR# upon detecting a target abort in response to an North Bridge initiated PCI transaction, upon being forced to end a non-configuration space transaction with a master abort, or if a parity error on the PCI bus is detected. If this bit is set to '0', the North Bridge will not assert SERR#.
Bit 7	AD	Address/Data stepping enable. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 6	P	PERR# response. Must always be set to '0'.
Bit 5	VGA	VGA Palette Snoop enable. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 4	MW	Master Write and Invalidate Enable. This bit is hardwired to a '0'. Writes to it have have no effect.
Bit 3	ES	Enable Special cycles. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 2	BS	Bus Master enabled. This bit is hardwired to a '1'. Writes to it have no effect.
Bit 1	ME	Memory Enable. This bit is hardwired to a '1'. Writes to it have no effect.
Bit 0	I/O E	I/O Enable. This bit is hardwired to a '1'. Writes to it have no effect.



8.5.2. NORTH BRIDGE PCI STATUS REGISTER

This is the 16-bit PCI Status register.

NB_Stat					Access = 0xCF8h/0xCFCh					Regoffset = 0x6h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DP	SS	SMA	RTA	STA	DT		DPED	FBBC	Rsv						
Default value after reset = 0280h															

Bit Number	Mnemonic	Description
Bit 15	DP	Detected parity error. This bit is set when a PCI parity error is detected. It may be cleared by software by writing a '1' back to this bit.
Bit 14	SS	Signalled SERR#. This bit is set to a '1' when SERR# is asserted by the North Bridge. Writing a '1' to this bit will clear it.
Bit 13	SMA	Signalled Master Abort. This bit is set to a 1 when the North Bridge terminates a PCI transaction with a master abort. Writing a '1' to this bit will clear it.
Bit 12	RTA	Received Target Abort. This bit is set to a '1' when PCI transaction initiated by the North Bridge is terminated with a target abort. Writing a '1' to this bit will clear it.
Bit 11	STA	Signalled Target Abort. This bit is hardwired to '0'.
Bits 10-9	DT	DEVSEL Timing. These bits are hardwired for medium timing to '01'. Writes have no effect.
Bit 8	DPED	Data Parity Error Detected. This bit is set to '1' when a PCI data parity error is detected. Writing a '1' will clear it.
Bit 7	FBBC	Fast Back-to-Back Capable. Hardwired to '1'. Indicates that the North Bridge, while acting as target, is capable of accepting fast back-to-back transactions. Reads will always return '1', writes have no effect.
Bits 6-0	Rsv	Reserved. These bits are hardwired to '0's.



8.5.3. NORTH BRIDGE PCI REVISION ID REGISTER

This is the 8-bit read only PCI revision identification register.

NB_R_ID		Access = 0xCF8h/0xCFCh				Regoffset = 0x8h	
7	6	5	4	3	2	1	0
Rsv							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0	Rsv	Reserved. These bits are hardwired to 00h.



8.5.4. NORTH BRIDGE DEVICE CLASS CODE REGISTER

This is a 24-bit read only register implemented at configuration space offset 9h, Ah, Bh.

NB_C_Code								Access = 0xCF8h/0xCFCh								Regoffset = 0x9h							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
BCC								SCC															
Default value after reset =								Default value after reset = 00h															

15	14	13	12	11	10	9	8								
PIR															
Default value after reset = 00h															

Bit Number	Mnemonic	Description
Bits 31-24	BCC	Base Class Code. These bits are hardwired to.
Bits 23-16	SCC	Sub Class Code. These bits are hardwired to 00h.
Bits 15-8	PIR	Programming Interface Register. These bits are hardwired to 00h.



8.5.5. NORTH BRIDGE HEADER TYPE REGISTER

This is an 8-bit read only register, hardwired to 00h

NB_Head

Access = 0xCF8h/0xCFCh

Regoffset = 0xEh

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0



8.5.6. NORTH BRIDGE CONTROL REGISTER

NB_Cont									Access = 0xCF8h/0xCFCh				Regoffset = 0x50h			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Rsv									PCI1	PCI2	PCI3	Rsv				
Default value after reset = 00000000h																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv											P			SP	S
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-23	Rsv	Reserved. Hardwired to '0'.
Bit 22	PCI1	PCI 2.0 Enable. If this bit is set to '1', North Bridge will be compatible with PCI 2.0 standard. If this bit is set to '0', North Bridge is compatible with PCI 2.1 standard.
Bit 21	PCI2	PCI to Host Read Prefetch Enable. If this bit is set to '1', all QWORD aligned burst reads from a PCI master addressed to the North Bridge system memory will use prefetch. If set to '0', memory read cycles from PCI to host are allowed to complete before the PCI cycle is terminated and all burst read attempts will be disconnected on the PCI bus.
Bit 20	PCI3	PCI to Host Write Posting Enable. If this bit is set to '1', all burst writes from a PCI master addressed to the North Bridge system memory will be posted. If it is set to '0', all memory write cycles from PCI to host are allowed to complete before the PCI cycle is terminated and all burst write attempts will be disconnected on the PCI bus.
Bits 19-5	Rsv	Reserved. Hardwired to '0'.
Bit 4	P	PERR_ on read data parity error enable.
Bit 3	P	PERR_ on write data parity error enable.
Bit 2	P	PERR_ on address parity error enable.
Bit 1	SP	SERR_ on PERR_ enable.
Bit 0	S	SERR_ on received target abort.



8.5.7. NORTH BRIDGE PCI ERROR STATUS REGISTER

NB_E_Stat					Access = 0xCF8h/0xCFCh							Regoffset = 0x54h			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv											RDP	WDP	AP	PES	RTAE
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-5	Rsv	Reserved. Hardwired to '0'.
Bit 4	RDP	Read Data Parity Error Status. This bit is set when a PCI read data parity error is detected. Writing a '1' will clear it.
Bit 3	WDP	Write Data Parity Error Status. This bit is set when a PCI write data parity error is detected. Writing a '1' will clear it.
Bit 2	AP	Address Parity Error Status. This bit is set when a PCI address parity error is detected. Writing a '1' will clear it.
Bit 1	PES	Parity Error Status. System errors as a result of a parity error status. This bit is set to '1' when SERR# was asserted as a result of parity error. Writing a '1' will clear it.
Bit 0	RTAE	Received Target Abort Error. System errors as a result of a received target abort. This bit is set to '1' when SERR# was asserted as a result of receiving a target abort. Writing a '1' will clear it.



8.6. THE SOUTH BRIDGE

The STPC South Bridge configuration registers are accessed using the values below :

- Bus = 0
- Device = Ch (IDSEL internally connected to PCI address line 12)
- Function = 0 (ISA bridge)
- Responds to I/O / memory / config

- Translates Master ISA to PCI

- Translates PCI to Slave ISA

8.7. SOUTH BRIDGE PCI ISA CONFIGURATION REGISTERS

Table 8-4. ISA Bridge Configuration Space Register Reset Values

31		16 15		0			
Device ID: 0210h			Vendor ID: 104Ah			00h	
Status: 0280h			Command: 000Fh			04h	
Base class code: 06h		Sub class code: 01h	Program. Inter. Reg. : 00h		Revision ID: 00h	08h	
		Header:					0Ch
							...
							...
					Miscellaneous reg : 00h		40h

This section describes Function 0 (F#0) configuration registers, including the PCI to ISA bridge control. The registers and reset values are illustrated in [Table 8-4](#).



8.7.1. SOUTH BRIDGE PCI COMMAND REGISTER

This is the 16-bit PCI command register.

SB_Com_0																Access = 0xCF8h/0xCFCh				Regoffset = 0x4h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Rsv							S	AD	P	VGA	MW	ESC	BM	ME	IO E								
Default value after reset = 000Fh																							

Bit Number	Mnemonic	Description
Bits 15-9	Rsv	Reserved. These bits are hardwired to '0's. Writes have no effect on them.
Bit 8	S	SERR# enable. If this bit is set to a '1', the South Bridge may assert SERR# upon detecting a target abort in response to a South Bridge initiated PCI transaction on behalf of an ISA master. If this bit is set to '0', the South Bridge will not assert SERR#.
Bit 7	AD	Address/Data stepping enable. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 6	P	PERR# response. Setting this bit to '1' enables parity error detection.
Bit 5	VGA	VGA Palette Snoop enable. This bits is hardwired to a '0'. Writes to it have no effect.
Bit 4	MW	Master Write and Invalidate Enable. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 3	ESC	Enable Special Cycles. This bit is hardwired to a '1'. The South Bridge writes to it have no effect. The South Bridge responds to halt and shutdown cycles.
Bit 2	BM	Bus Master enabled. This bit is hardwired to a '1'. Writes to it have no effect.
Bit 1	ME	Mem Enable. This bit is hardwired to a '1'. Writes to it have no effect.
Bit 0	I/O E	I/O Enable. This bit is hardwired to a '1'. Writes to it have no effect.



8.7.2. SOUTH BRIDGE PCI STATUS REGISTER

This is the 16-bit PCI Status register.

SB_Stat0					Access = 0xCF8h/0xCFCh					Regoffset = 0x6h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv	SS	SMA	RTA	STA	DT		DPED	FBBC	Rsv						
Default value after reset = 0280h															

Bit Number	Mnemonic	Description
Bit 15	Rsv	Reserved. This bit is hardwired to '0'.
Bit 14	SS	Signalled SERR#. This bit is set to a '1' when SERR# is asserted by the South Bridge on behalf of an ISA master cycle. Writing a '1' to this bit will clear it. SERR# is asserted in response to a target abort during an ISA master cycle on PCI bus and if bit-8 of the F#0 PCI command register is set to a '1' to enable SERR# signalling.
Bit 13	SMA	Signalled Master Abort. This bit is hardwired to a '0'.
Bit 12	RTA	Received Target Abort. This bit is set to a '1' when the PCI transaction is initiated by the South Bridge on behalf of an ISA master is terminated with a target abort. Writing a '1' to this bit will clear it.
Bit 11	STA	Signalled Target Abort. This bit is set to a '1' when the South Bridge terminates a PCI transaction with a target abort. Writing a '1' to this bit will clear it. The South Bridge will generate target abort if a A1-0 of a PCI IO cycle does not match the Byte enables.
Bits 10-9	DT	DEVSEL Timing. These bits are hardwired for medium timing to '01'. Writes have no effect.
Bit 8	DPED	Data Parity Error Detected. This bit is hardwired to '0'.
Bit 7	FBBC	Fast Back-to-Back Capable. Hardwired to '1'. Indicates that the South Bridge, while acting as target, is capable of accepting fast back-to-back transactions. Reads will always return '1', writes have no effect.
Bits 6-0	Rsv	Reserved. These bits are hardwired to '0'.



8.7.3. SOUTH BRIDGE PCI REVISION ID REGISTER

This is the 8-bit read only PCI revision identification register.

SB_R_ID0		Access = 0xCF8h/0xCFCh				Regoffset = 0x8h	
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Bits 7-0: These bits are hardwired to 00h.



8.7.4. SOUTH BRIDGE DEVICE CLASS CODE REGISTER

This is a 24-bit read only register implemented at configuration space offset 09h, 0Ah, 0Bh.

SB_C_Code0								Access = 0xCF8h/0xCFCh								Regoffset = 0x9h							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
BCC								SCC															
Default value after reset = 06h								Default value after reset = 01h															

15	14	13	12	11	10	9	8								
PIR															
Default value after reset = 00h															

Bit Number	Mnemonic	Description
Bits 31-24	BCC	Base Class Code. These bits are hardwired to 06h (Bridge Device).
Bits 23-16	SCC	Sub Class Code. These bits are hardwired to 01h (ISA Bridge).
Bits 15-8	PIR	Programming Interface Register. These bits are hardwired to 00h.



8.7.5. SOUTH BRIDGE HEADER TYPE REGISTER

This register is hardwired to 80h, indicating that the South Bridge is a multi-function PCI device.

SB_Head0		Access = 0xCF8h/0xCFCh				Regoffset = 0xEh	
7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0



8.7.6. SOUTH BRIDGE MISCELLANEOUS REGISTER

SB_Misc0				Access = 0xCF8h/0xCFCh			Regoffset = 040h
7	6	5	4	3	2	1	0
Rsv				GPIO_MDC	GPIO_SDC	GPIO	PCI
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-4	Rsv	Reserved. Hardwired to 00h.
Bit 3	GPIO_MDC	GPIO Master Debounce Control : enable/disable the debounce logic on master GPIO 0 = Disabled 1 = Enabled
Bit 2	GPIO_SDC	GPIO Slave Debounce Control : enable/disable the debounce logic on slave GPIO 0 = Disabled 1 = Enabled
Bits 1	GPIO	GPIO ¹⁾ Enable/Disable: enable/disable all GPIOs 0 = Disabled 1 = Enabled
Bit 0	PCI	PCI 2.0 Enable . If this bit is set to '1', South Bridge will be compatible with PCI 2.0 standard. If this bit is set to '0', South Bridge is compatible with PCI 2.1 standard.

Note 1. Enabling GPIO will enable both master and slave GPIO controllers.

Programming Notes; This register is used to initialise the GPIOs as well as select the mode in which the Bridge operates. For information on how to initialise the GPIOs, refer to [Section 12](#).



8.8. PCI to IDE BRIDGE CONFIGURATION REGISTERS

This section describes the PCI to IDE Bridge configuration registers. The registers and reset values are illustrated in [Table 8-5](#).

The PCI to IDE Bridge configuration registers are accessed using the values below:

Bus = 0

Device = 0Dh (IDSEL internally connected to PCI address line 13)

Function = 0(IDE controller)

- Responds to IO / config

For example: Writing 80006800h at CF8h will access Function 0 (IDE) Command reg. index.

Table 8-5. PCI to IDE Bridge Configuration Space Register Reset Values

31	16 15			0
Device: 0229h		Vendor ID: 104Ah		00h
Status: 0280h		Command: 0000h		04h
Base class code: 01h	Sub class code: 01h	Program. Inter. Reg: 8Ah	Revision ID: 00h	08h
	Header: 40h	Reserved: 00h		0Ch
IO Base address 0 register: 00000001h				10h
IO Base address 1 register: 00000001h				14h
IO Base address 2 register: 00000001h				18h
IO Base address 3 register: 00000001h				1Ch
Reserved				20h
				...
				...
MISC REGISTER 0				40h
				44h
MISC REGISTER 1				48h
				4Ch
PRIMARY MASTER TIMING REGISTER				50h
PRIMARY SLAVE TIMING REGISTER				54h
SECONDARY MASTER TIMING REGISTER				58h
SECONDARY SLAVE TIMING REGISTER				5Ch



PCI CONTROLLERS

8.8.1. PCI to IDE BRIDGE PCI COMMAND REGISTER

This is the 16-bit PCI command register.

IDEB_Com1							Access = 0xCF8h/0xCFCh					Regoffset = 0x4h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv							SE	A	P	VGA	MWIE	ESC	BM	M E	IO E
Default value after reset = 0005h															

Bit Number	Mnemonic	Description
Bits 15-9	Rsv	Reserved. These bits are hardwired to '0's. Writes have no effect on them.
Bit 8	SE	SERR# Enable. If this bit is set to a '1', the PCI to IDE Bridge may assert SERR# upon detecting a master or target abort in response to a the PCI to IDE Bridge initiated PCI transaction on behalf of IDE master. If this bit is set to '0', the PCI to IDE Bridge will not assert SERR#.
Bit 7	A	Address/Data stepping enable. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 6	P	PERR# response. Setting this bit to '1' enables parity error detection.
Bit 5	VGA	VGA Palette Snoop enable. This bits is hardwired to a '0'. Writes to it have no effect.
Bit 4	MWIE	Master Write and Invalidate Enable. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 3	ESC	Enable Special cycles. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 2	BM	Bus Master enabled. This bit is hardwired to a '1'. Writes to it have no effect.
Bit 1	M E	Mem Enable. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 0	IO E	IO Enable. Setting this bit to a '1' enables access to the IDE IO registers.



8.8.2. PCI to IDE BRIDGE PCI STATUS REGISTER

IDEB_Stat1										Access = 0xCF8h/0xCFCh					Regoffset = 0x6h				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Rsv	SS	SMA	RTA	Rsv	DT		DPED	FBBC	Rsv										
Default value after reset = 0280h																			

Bit Number	Mnemonic	Description
Bit 15	Rsv	Reserved. This bit is hardwired to '0'.
Bit 14	SS	Signalled SERR#. This bit is set to a '1'.
Bit 13	SMA	Signalled Master Abort. This bit is set to a '1' when the PCI to IDE bridge terminates a PCI transaction initiated on behalf of the IDE master with a master abort. The PCI to IDE bridge master aborts an IDE master cycle if no target responds to this cycle.
Bit 12	RTA	Received Target Abort. This bit is set to a '1' when a PCI transaction initiated by the PCI to IDE bridge on behalf of the IDE master is terminated with a target abort. Writing a '1' to this bit will clear it.
Bit 11	Rsv	Reserved. This bit is hardwired to '0'.
Bits 10-9	DT	DEVSEL Timing. These bits are hardwired for medium timing to '01'. Writes to these bits have no effect.
Bit 8	DPED	Data Parity Error Detected. This bit is hardwired to '0'.
Bit 7	FBBC	Fast Back-to-Back Capable. This bit is hardwired to '1'.
Bits 6-0	Rsv	Reserved. These bits are hardwired to '0'.



8.8.3. PCI to IDE BRIDGE REVISION ID REGISTER

This is the 8-bit read only PCI revision identification register.

IDEB_R_ID1		Access = 0xCF8h/0xCFCh				Regoffset = 0x8h	
7	6	5	4	3	2	1	0
Rsv							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0	Rsv	Reserved. These bits are hardwired to 00h in this stepping of the chip.



8.8.4. PCI to IDE BRIDGE PROGRAMMING INTERFACE REGISTER

Prog_Int				Access = 0xCF8h/0xCFCh			Regoffset = 0x09h	
7	6	5	4	3	2	1	0	
Rsv	Rsv			SCPS	OPMODE2	SCPP	OPMODE1	
Default value after reset = 8Ah								

Bit Number	Mnemonic	Description
Bit 7	Rsv	Reserved. This bit is hardwired to '1'. Writes to have no effect on this bit.
Bits 6-4	Rsv	Reserved. These bits are hardwired to '0'. These bits are hardwired to '0'.
Bit 3	SCPS	This bit is hardwired to '1' indicating that the secondary channel is programmable to be either in legacy or native mode.
Bit 2	OPMODE2	This bit selects the operating mode of the secondary channel. (see Table 8-6)
Bit 1	SCPP	This bit is hardwired to '1' indicating that the primary channel is programmable to be either in legacy or native mode.
Bit 0	OPMODE1	This bit selects the operating mode of the primary channel. (see Table 8-7)

Table 8-6. Operating Mode of the Secondary Channel

Bit 2	
0	Channel is in legacy mode. In legacy mode the secondary IDE channel occupies IO addresses 170h-177h and 376h.
1	Channel is in native mode. The address range occupied by the secondary IDE controller in native mode is specified by base address registers 2 and 3.

Table 8-7. Operating Mode of the Primary Channel

Bit 0	
0	Channel is in legacy mode. In legacy mode the Primary IDE channel occupies IO addresses 1F0h-1F7h and 3F6h.
1	Channel is in native mode. The address range occupied by the Primary IDE controller in native mode is specified by base address registers '0' and '1'.



8.8.5. PCI to IDE BRIDGE SUB-CLASS CODE REGISTER

This register is hardwired to 01h indicating that this is an IDE controller device.

Sub_Class				Access = 0xCF8h/0xCFCh				Regoffset = 0xAh			
7	6	5	4	3	2	1	0				
0	0	0	0	0	0	0	1				



8.8.6. PCI to IDE BRIDGE BASE-CLASS CODE REGISTER

This register is hardwired to 01h indicating that Function 1 is a mass storage device.

Base_Class		Access = 0xCF8h/0xCFCh				Regoffset = 0xBh	
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1



8.8.7. PCI to IDE BRIDGE LATENCY TIMER CONTROL REGISTER

Lat_T		Access = 0xCF8h/0xCFCh				Regoffset = 0xDh	
7	6	5	4	3	2	1	0
Rsv							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0	Rsv	Reserved. These bits are hardwired to '0'.



8.8.8. PCI to IDE BRIDGE HEADER TYPE REGISTER

This register is hardwired to 40h indicating that the PCI to IDE Bridge is a PCI multi-function device.

<i>Head_T</i>							
Access = 0xCF8h/0xCFCh				Regoffset = 0xEh			
7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	0



8.8.9. PCI to IDE BRIDGE IDE BASE ADDRESS 0 REGISTER

This 32-bit register contains the base IO address for accessing the primary IDE channel's command registers. The base address is meaningful only when the Primary channel is programmed for native mode operation. If programmed for legacy mode operation, the primary channel's command registers are decoded at 1F0h IO address.

Base0				Access = 0xCF8h/0xCFCh								Regoffset = 0x10h			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA															
Default value after reset = 00000001h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA													Rsv	Rsv	MSI
Default value after reset = 00000001h															

Bit Number	Mnemonic	Description
Bits 31-3	BA	Base Address. This field specifies the 8-Byte IO address range where the primary channel command registers are located.
Bit 2	Rsv	Hardwired to '0' to indicate that this base address occupies 4-Bytes in IO space.
Bit 1	Rsv	Reserved. Hardwired to '0'.
Bit 0	MSI	Memory Space Indicator. This bit is hardwired to '1' to indicate IO space.



8.8.10. PCI to IDE BRIDGE IDE BASE ADDRESS 1 REGISTER

This 32-bit register contains the base IO address for accessing the primary IDE channel's Control registers. The base address is meaningful only when the Primary channel is programmed for native mode operation. If programmed for legacy mode operation, the primary channel's control register are decoded at 3F6h.

Base1 Access = 0xCF8h/0xCFCh Regoffset = 0x14h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA															
Default value after reset = 00000001h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA														Rsv	MSI
Default value after reset = 00000001h															

Bit Number	Mnemonic	Description
Bits 31-2	BA	Base Address. This field specifies the 4-Byte IO address range where the primary channel command registers are located.
Bit 1	Rsv	Reserved. Hardwired to '0'.
Bit 0	MSI	Memory Space Indicator. This bit is hardwired to '1' to indicate IO space.



8.8.11. PCI to IDE BRIDGE IDE BASE ADDRESS 2 REGISTER

This 32-bit register contains the base IO address for accessing the secondary IDE channel's command registers. The base address is meaningful only when the secondary channel is programmed for native mode operation. If programmed for legacy mode operation, the secondary channel's command registers are decoded at 170h IO address.

Base2				Access = 0xCF8h/0xCFCh												Regoffset = 0x18h			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
BA																			
Default value after reset = 00000001h																			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA													Rsv	Rsv	MSI
Default value after reset = 00000001h															

Bit Number	Mnemonic	Description
Bits 31-3	BA	Base Address. This field specifies the 8-Byte IO address range where the secondary channel command registers are located.
Bit 2	Rsv	Hardwired to '0' to indicate that this base address occupies 4-Bytes in IO space.
Bit 1	Rsv	Reserved. Hardwired to '0'.
Bit 0	MSI	Memory Space Indicator. This bit is hardwired to '1' to indicate IO space.



8.8.12. PCI to IDE BRIDGE IDE BASE ADDRESS 3 REGISTER

This 32-bit register contains the base IO address for accessing the secondary IDE channel's Control registers. The base address is meaningful only when the secondary channel is programmed for native mode operation. If programmed for legacy mode operation, the secondary channel's control register is decoded at 376h.

Base3				Access = 0xCF8h/0xCFCh								Regoffset = 0x1Ch			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA															
Default value after reset = 00000001h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA														Rsv	MSI
Default value after reset = 00000001h															

Bit Number	Mnemonic	Description
Bits 31-2	BA	Base Address. This field specifies the 4-Byte IO address range where the secondary channel command registers are located.
Bit 1	Rsv	Reserved. Hardwired to '0'.
Bit 0	MSI	Memory Space Indicator. This bit is hardwired to '1' to indicate IO space.



8.8.13. PCI to IDE BRIDGE IDE BASE ADDRESS 4 REGISTER

This 32-bit register contains the base IO address for accessing the bus master control and status register.

Base4				Access = 0xCF8h/0xCFCh								Regoffset = 0x20h							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
BA																			
Default value after reset = 00000001h																			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												HW		Rsv	MSI
Default value after reset = 00000001h															

Bit Number	Mnemonic	Description
Bits 31-4	BA	Base Address. This field specifies the 16-bytes IO address range where the Bus master control and status registers are located.
Bits 3-2	HW	Hardwired to '0' to indicate that this base address occupies 16-bytes in IO space.
Bit 1	Rsv	Reserved. Hardwired to '0'.
Bit 0	MSI	Memory space indicator. This bit is hardwired to '1' to indicate IO space.



8.8.14. MISC REGISTER 0

MISC_REG0

Access = 0xCF8h/0xCFCh

Regoffset = 0x40h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSMC				SMMC				PSMC				PMMC			
Reset value: 0x00h															

Bit Number	Mnemonic	Description
Bits 15-12	SSMC	Secondary Slave Misc Control Bits: 1 = Enable; 0 = Disable Bit 15: IOCHRDY Sampling enable/disable Bit 14: Write Posting enable/disable Bit 13: Read Prefetch enable/disable Bit 12: ATAPI Command Prefetch enable/disable
Bits 11-8	SMMC	Secondary Master Misc Control Bits: 1 = Enable; 0 = Disable Bit 11: IOCHRDY Sampling enable/disable Bit 10: Write Posting enable/disable Bit 9: Read Prefetch enable/disable Bit 8: ATAPI Command Prefetch enable/disable
Bits 7-4	PSMC	Primary Slave Misc Control Bits: 1 = Enable; 0 = Disable Bit 7: IOCHRDY Sampling enable/disable Bit 6: Write Posting enable/disable Bit 5: Read Prefetch enable/disable Bit 4: ATAPI Command Prefetch enable/disable
Bits 3-0	PMMC	Primary Master Misc Control Bits: 1 = Enable; 0 = Disable Bit 3: IOCHRDY Sampling enable/disable Bit 2: Write Posting enable/disable Bit 1: Read Prefetch enable/disable Bit 0: ATAPI Command Prefetch enable/disable

8.8.15. MISC REGISTER 1

MISC_REG1

Access = 0xCF8h/0xCFChh

Regoffset = 0x48h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv				SME	SSE	PME	PSE	ISR	Rsv			SD	PD	SC	PC
Default value after reset = 0x00h															

Bit Number	Mnemonic	Description
Bits 31-12	Rsv	Reserved: read 0.
Bit 11	SSE	Secondary Slave UDMA Enable: 1 = Enable; 0 = Disable
Bit 10	SME	Secondary Master UDMA Enable: 1 = Enable; 0 = Disable
Bit 9	PSE	Primary Slave UDMA Enable: 1 = Enable; 0 = Disable
Bit 8	PME	Primary Master UDMA Enable: 1 = Enable; 0 = Disable
Bit 7	ISR	IDE Soft Reset: 1 = Reset
Bits 6-4	Rsv	Reserved: read 0.
Bit 3	SD	Secondary Disable:
Bit 2	PD	Primary Disable:
Bit 1	SC	Secondary Config: s_cfg_intr_r
Bit 0	PC	Primary Config: p_cfg_intr_r



8.8.16. PRIMARY MASTER TIMING REGISTER

PMT_REG Access = 0xCF8h/0xCFCh Regoffset = 0x50h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMA_C					DMA_ATC						DMA_R				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_R	PIO_ADD			PIO_A						PIO_R					
Default value after reset = 0X1C854D3h															

Bit Number	Mnemonic	Description
Bits 31-27	UDMA_C	UDMA Time: number of clocks.
Bits 26-21	DMA_ATC	DMA Active Time: number of clocks.
Bits 20-15	DMA_R	DMA Recovery Time: number of clocks.
Bits 14-12	PIO_ADD	PIO Address Setup: number of clocks.
Bits 11-6	PIO_A	PIO Active: number of clocks.
Bits 5-0	PIO_R	PIO Recovery: number of clocks.



8.8.17. PRIMARY SLAVE TIMING REGISTER

PST_REG

Access = 0xCF8h/0xCFCh

Regoffset = 0x54h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMA_C					DMA_ATC						DMA_R				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_R	PIO_ADD			PIO_A						PIO_R					
Default value after reset = 0X1C854D3h															

Bit Number	Mnemonic	Description
Bits 31-27	UDMA_C	UDMA Time: number of clocks.
Bits 26-21	DMA_ATC	DMA Active Time: number of clocks.
Bits 20-15	DMA_R	DMA Recovery Time: number of clocks.
Bits 14-12	PIO_ADD	PIO Address Setup: number of clocks.
Bits 11-6	PIO_A	PIO Active: number of clocks.
Bits 5-0	PIO_R	PIO Recovery: number of clocks.



8.8.18. SECONDARY MASTER TIMING REGISTER

SMT_REG Access = 0xCF8h/0xCFCh Regoffset = 0x58h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMA_C					DMA_ATC						DMA_R				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_R	PIO_ADD			PIO_A						PIO_R					
Default value after reset = 0X1C854D3h															

Bit Number	Mnemonic	Description
Bits 31-27	UDMA_C	UDMA Time: number of clocks.
Bits 26-21	DMA_ATC	DMA Active Time: number of clocks.
Bits 20-15	DMA_R	DMA Recovery Time: number of clocks.
Bits 14-12	PIO_ADD	PIO Address Setup: number of clocks.
Bits 11-6	PIO_A	PIO Active: number of clocks.
Bits 5-0	PIO_R	PIO Recovery: number of clocks.



PCI CONTROLLERS

8.8.19. SECONDARY SLAVE TIMING REGISTER

SST_REG Access = 0xCF8h/0xCFCh Regoffset = 0x5Ch

Access = 0xCF8h/0xCFCh Regoffset = 0x5Ch

Regoffset = 0x5Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDMA_C					DMA_ATC						DMA_R				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_R	PIO_ADD			PIO_A						PIO_R					
Default value after reset = 0x1C854D3h															

Bit Number	Mnemonic	Description
Bits 31-27	UDMA_C	UDMA Time: number of clocks.
Bits 26-21	DMA_ATC	DMA Active Time: number of clocks.
Bits 20-15	DMA_R	DMA Recovery Time: number of clocks.
Bits 14-12	PIO_ADD	PIO Address Setup: number of clocks.
Bits 11-6	PIO_A	PIO Active: number of clocks.
Bits 5-0	PIO_R	PIO Recovery: number of clocks.

8.9. PCI to USB BRIDGE CONFIGURATION REGISTERS

The PCI to USB Bridge configuration registers are accessed using the values below:

Bus = 0

Device = 0Eh (IDSEL internally connected to PCI address line 14)

Function = 0 (USB controller)

- Responds to IO / config

For example: Writing 80007000h at CF8h will access Function 0 (USB) Command reg. index 0h.

Table 8-8. PCI to USB Bridge Configuration Space Register Reset Values

31	16 15				0
Device: 0230h			Vendor ID: 104Ah		00h
Status: 0280h			Command: 0000h		04h
Base class code: 0Ch		Sub class code: 03h	Program. Inter. Reg.: 10h	Revision ID: 00h	08h
		Header: 40h	Reserved: 00h		0Ch
Command					10h
Class_Code					14h
BAR_OHCI					18h
					1Ch
					20h



8.9.1. PCI to USB BRIDGE VENDOR IDENTIFICATION REGISTER

This is a 16-bit read-only register implemented at configuration space offset 00h and 01h. It contains the Vendor Identifier assigned to STPC.

Bits 15-0 These bits are hardwired to 104Ah.

USBB_V_ID1

Access = 0xCF8h/0xCFCh

Regoffset = 0x0h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	0

Writes to this register have no effect.



8.9.2. PCI to USB BRIDGE DEVICE IDENTIFICATION REGISTER

This is a 16-bit read only register implemented at configuration space offset 02h and 03h. It contains the Device Identifier assigned to the PCI to USB bridge.

Bits 15-0 These bits are hardwired to 0230h

USBB_D_ID1

Access = 0xCF8h/0xCFCh

Regoffset = 0x2h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0

Writes to this register have no effect.



8.9.3. USB BRIDGE PCI COMMAND REGISTER

This is the 16-bit PCI command register.

USBB_Com							Access = 0xCF8h/0xCFCh					Regoffset = 0x4h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv							S	AD	P	VGA	MW	ES	BS	ME	IO E
Default value after reset = 0006h															

Bit Number	Mnemonic	Description
Bits 15-9	Rsv	Reserved. These bits are hardwired to '0'. Writes have no effect on them.
Bit 8	S	SERR# enable. If this bit is set to a '1', the USB Bridge may assert SERR# upon detecting a target abort in response to a USB Bridge initiated PCI transaction, upon being forced to end a non-configuration space transaction with a master abort, or if a parity error on the PCI bus is detected. If this bit is set to '0', the USB Bridge will not assert SERR#.
Bit 7	AD	Address/Data stepping enable. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 6	P	PERR# response. Must always be set to '0'.
Bit 5	VGA	VGA Palette Snoop enable. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 4	MW	Master Write and Invalidate Enable. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 3	ES	Enable Special cycles. This bit is hardwired to a '0'. Writes to it have no effect.
Bit 2	BS	Bus Master enabled. This bit is hardwired to a '1'. Writes to it have no effect.
Bit 1	ME	Memory Enable. This bit is hardwired to a '1'. Writes to it have no effect.
Bit 0	I/O E	I/O Enable. This bit is hardwired to a '0'. Writes to it have no effect.



8.9.4. USB BRIDGE PCI STATUS REGISTER

This is the 16-bit PCI Status register.

USBB_Stat

Access = 0xCF8h/0xCFCh

Regoffset = 0x6h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DP	SS	SMA	RTA	STA	DT		DPED	FBBC	Rsv						
Default value after reset = 0280h															

Bit Number	Mnemonic	Description
Bit 15	DP	Detected parity error. This bit is set when a PCI parity error is detected. It may be cleared by software by writing a '1' back to this bit.
Bit 14	SS	Signalled SERR#. This bit is set to a '1' when SERR# is asserted by the USB Bridge. Writing a '1' to this bit will clear it.
Bit 13	SMA	Signalled Master Abort. This bit is set to a 1 when the USB Bridge terminates a PCI transaction with a master abort. Writing a '1' to this bit will clear it.
Bit 12	RTA	Received Target Abort. This bit is set to a '1' when PCI transaction initiated by the USB Bridge is terminated with a target abort. Writing a '1' to this bit will clear it.
Bit 11	STA	Signalled Target Abort. This bit is hardwired to '0'.
Bits 10-9	DT	DEVSEL Timing. These bits are hardwired for medium timing to '01'. Writes have no effect.
Bit 8	DPED	Data Parity Error Detected. This bit is set to '1' when a PCI data parity error is detected. Writing a '1' will clear it.
Bit 7	FBBC	Fast Back-to-Back Capable. Hardwired to '1'. Indicates that the USB Bridge, while acting as target, is capable of accepting fast back-to-back transactions. Reads will always return '1', writes have no effect.
Bits 6-0	Rsv	Reserved. These bits are hardwired to '0's.

8.9.5. USB BRIDGE PCI REVISION ID REGISTER

This is the 8-bit read only PCI revision identification register.

USBB_R_ID		Access = 0xCF8h/0xCFCh				Regoffset = 0x8h	
7	6	5	4	3	2	1	0
Rsv							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0	Rsv	Reserved. These bits are hardwired to 00h.



8.9.6. USB BRIDGE DEVICE CLASS CODE REGISTER

This is a 24-bit read only register implemented at configuration space offset 9h, Ah, Bh.

USBB_C_Code								Access = 0xCF8h/0xCFCh								Regoffset = 0x9h							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
BCC								SCC															
Default value after reset = 0Ch								Default value after reset = 03h															

15	14	13	12	11	10	9	8								
PIR															
Default value after reset = 10h															

Bit Number	Mnemonic	Description
Bits 31-24	BCC	Base Class Code. These bits are hardwired to 0Ch.
Bits 23-16	SCC	Sub Class Code. These bits are hardwired to 03h.
Bits 15-8	PIR	Programming Interface Register. These bits are hardwired to 10h.



8.9.7. USB BRIDGE HEADER TYPE REGISTER

This is an 8-bit read only register, hardwired to 40h

USBB_Head

Access = 0xCF8h/0xCFCh

Regoffset = 0xEh

7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	0



8.10. PCI Configuration for OpenHCI-compliant USB Host Controller

The following table provides a summary of the registers that are necessary for the USB Host Controller to be successfully configured in a PCI-based PC host. Those registers which are implementation-dependent are not described in the table; their implementation is left to the individual manufacturers for innovation. However, they are defined in the PCI Specification, Revision 2.1.

Register	Offset	Description
Command	05 - 04	Provides coarse control over a device's ability to generate and respond to PCI cycles
CLASS_CODE	0B - 09	Identifies the generic function of the device
BAR_OHCI	13 - 10	Specifies the base address of a contiguous block in the main memory of the PC host, from which 4 KB of directly-mapped addressing spaces are reserved by OpenHCI for the operational registers of the Host Controller



8.10.1. COMMAND REGISTER

This register provides coarse control over the ability to generate and respond to PCI cycles. It is imperative that the Host Controller supports both PCI bus-mastering and memory-mapping of all operational registers into the main memory of the PC host. Consequently, the fields **MA** and **BM** should always be set to '1' during device configuration.

Once the Host Controller has started processing endpoint lists of periodic and nonperiodic, the action to reset either field **MA** or **BM** to its default value should be approached with caution. If the field **MA** is reset to '0', the Host Controller can no longer respond to any software command addressed to it and interrupt generation is halted, while the Host Controller can still generate the SOF token at the beginning of each frame. If the field **BM** is reset to '0', the Host Controller will no longer be able to read Distributors (both Endpoint and Transfer) from the main memory, nor can it update the *HCCA* partition in the main memory.

Command																Access = h				Regoffset = 0xh					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Rsv									
Default value after reset = 0																									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*							Rsv						BM	MA	*
Default value after reset = 0															

Bit Number	Mnemonic	Description
Bits 31 - 16	Rsv	Reserved
Bits 15 - 9	*	Refer to PCI Specification, Revision 2.1, for definition
Bits 8 - 3	Rsv	Reserved
Bit 2	BM	BUS MASTER Default '0' Indicates the device's ability to act as a bus-master
Bit 1	MA	MEMORY ACCESS Default '0' Indicates the device's ability to respond to PCI memory cycles
Bit 0	*	Refer to PCI Specification, Revision 2.1, for definition



8.10.2. CLASS_CODE Register

This register identifies the basic function of the device, and a specific programming interface code for a USB Host Controller.

CLASS_CODE								Access = h				Regoffset = 0xh			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv								BC							
Default value after reset = 0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SC								PI							
Default value after reset = 0															

Bit Number	Mnemonic	Description
Bits 31 - 24	Rsv	Reserved
Bits 23 - 16	BC	BASE CLASS A constant value of '0Ch' Identifies the device being a Serial Bus Controller
Bits 15 - 8	SC	SUB CLASS A constant value of '03h' Identifies the device being of Universal Serial Bus
Bits 7 - 0	PI	PROGRAMMING INTERFACE A constant value of '10h' Identifies the device being a Host Controller



8.10.3. BAR_OHCI Register

The *BAR_OHCI* register specifies the base address of a contiguous memory space in the main memory of the PC host, which is reserved for the operational registers defined by the Specification, Release 1.0. All of the operational registers are directly mapped into this memory space. With reference to the PCI Specification, Revision 2.1, the Host Controller Driver will always allocate a memory band of 4 KB for the operational registers. This is despite the fact that the number of operational registers defined by the Specification, Release 1.0, is far less than 4 KB. Regardless of whether the hardware vendor of a USB Host Controller chooses to implement the decoding logic for bits [11:0] or not, the respective hardware **must** be able to decode the operational registers. When any of the addresses between the block of operational registers and the 4-KB upper-bound is accessed, the hardware is not required to respond and the access can be ignored.

The hardware registers that are used to implement vendor specific features are not covered by the Specification, Release 1.0. Consequently, any vendor-specific hardware registers **should not** be mapped into the memory space starting at the address location as indicated by *BAR_OHCI*.

<i>BAR_OHCI</i>															
Access = h															
Regoffset = 0xh															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BAR															
Default value after reset = 0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BAR												PM	TP		IND
Default value after reset = 0															

Bit Number	Mnemonic	Description
Bits 31 - 12	BAR	BASE ADDRESS Specifies the upper 20 bits of the 32-bit starting base address. This represents a maximum of 4-KB addressing space for the OpenHCI's operational registers
Bits 11 - 4		Default value of '00h' and is read only Represents a maximum of 4-KB addressing space for the OpenHCI's operational registers
Bit 3	PM	PREFETCH MEMORY A constant value of '0' Indicates that there is no support for "prefetchable memory"



Bit Number	Mnemonic	Description
Bits 2 - 1	TP	TYPE A constant value of '00b' Indicates that the base register is 32-bit wide and can be placed anywhere in the 32-bit memory space; i.e., lower 4 GB of the main memory of the PC host
Bit 0	IND	INDICATOR A constant value of '0b' Indicates that the operational registers of the device are mapped into memory space of the main memory of the PC host system



8.11. Legacy USB Support Registers

Four operational registers are used to provide USB legacy support. Each of these registers is located on a 32-bit boundary. The offset of these registers is relative to the base address of the Host Controller operational registers with *HceControl* located at offset 100h.

Register	Offset	Description
HceControl	100h	Used to enable and control the emulation hardware and report various status information.
HceInput	104h	Emulation side of the legacy Input Buffer register.
HceOutput	108h	Emulation side of the legacy Output Buffer register where keyboard and mouse data is to be written by software.
Hce Status	10Ch	Emulation side of the legacy Status register.



8.12. PCI to LAN BRIDGE CONFIGURATION REGISTERS

The PCI to LAN Bridge configuration registers are accessed using the values below:

Bus = 0

Device = 0Fh (IDSEL internally connected to PCI address line 15)

Function = 0 (LAN controller)

- Responds to IO / config

Table 8-9. PCI to LAN Bridge Configuration Space Register Reset Values

31		16 15		0		
Device: 0238h			Vendor ID: 104Ah			00h
Status: 0280h			Command: 0000h			04h
Base class code: 03h		Sub class code: 00h	Program. Inter. Reg. : 00h		Revision ID: 00h	08h
BIST: 00h		Header: 00h	Latency Timer: 00h		Cache Line Size: 00h	0Ch
Memory Base Address						10h
I/O Base Address						14h
BAR_OHCI						18h
						1Ch
						20h
			Interrupt Pin		Interrupt Line	3Ch

Note: The Ethernet - PCI Configuration Register (Function2) registers are listed in [Table 14-2](#).

PCI CONTROLLERS

8.12.1. Command Reg

COMMAND																Access = h				Regoffset = 0x04h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Rsv							SERR En	Rsv					BM En	Mem En	I/O En								
Default value after reset = 0x0000h																							

This is the 16-bit PCI command register.

Bit Number	Mnemonic	Description
Bits 15:10	Rsv	Reserved, hardwired to 0
Bit 9	Rsv	Reserved, hardwired to 0
Bit 8	SERR En	SERR# Enable.
Bit 7	Rsv	Reserved, hardwired to 0
Bit 6	Rsv	Reserved, hardwired to 0
Bit 5	Rsv	Reserved, hardwired to 0
Bit 4	Rsv	Reserved, hardwired to 0
Bit 3	Rsv	Reserved, hardwired to 0
Bit 2	BM En	Bus Master Enable, on the primary side: 0 = Bus Master is disabled 1 = Bus Master is enabled to act as master on the PCI bus
Bit 1	Mem En	Memory space Enable: 0 = Memory decode for this card disabled 1 = Memory decode for this card enabled.
Bit 0	I/O En	I/O space Enable: 0 = I/O decode disabled 1 = I/O decode enabled



8.12.2. Status Register

This is the 16-bit PCI Status register.

STATUS				Access = h		Regoffset = 0x06	
15	14	13	12	11	10	9	8
Rsv	SSE	RMA	RTA	Rsv			
Default value after reset = 0x0280h							

7	6	5	4	3	2	1	0
Rsv							
Default value after reset = 0x0280h							

Bit Number	Mnemonic	Description
Bit 15	Rsv	Reserved, hardwired to 0
Bit 14	SSE	Signalled System Error.
Bit 13	RMA	Received Master Abort.
Bit 12	RTA	Received Target Abort.
Bit 11	Rsv	Reserved, hardwired to 0
Bits 10:9	Rsv	Reserved, hardwired to 01
Bit 8	Rsv	Reserved, hardwired to 0
Bit 7	Rsv	Reserved, hardwired to 1
Bit 6	Rsv	Reserved, hardwired to 0
Bit 5	Rsv	Reserved, hardwired to 0
Bits 4:0	Rsv	Reserved, hardwired to 0.



PCI CONTROLLERS

8.12.3. Revision ID

This is the 8-bit read only PCI revision identification register.

REVISION_ID				Access = h		Regoffset = 0x08h	
7	6	5	4	3	2	1	0
Hardwired to 0x00							

8.12.4. Class Code

This is a 24-bit read only register.

CLASS_CODE						Access = h				Regoffset = 0x09h					
23	22	21	20	18	18	17	16	15	14	13	12	Rsv			
Default value after reset = 200h															

11	10	9	8	7	6	5	4	3	2	1	0
Rsv											
Default value after reset = 200h											

Bit Number	Mnemonic	Description
Bits 23:16	Rsv	Reserved Hardwired to 02h
Bits 15:8	Rsv	Reserved Hardwired to 00h
Bits 7:0	NPI	Reserved Hardwired to 00h



8.12.5. Cache Line Size

CACHE_LINE_SIZE			Access = h			Regoffset = 0x0Ch	
7	6	5	4	3	2	1	0
Hardwired to 0x00h							

8.12.6. Latency Timer

LATENCY_TIMER			Access = h			Regoffset = 0x0Dh	
7	6	5	4	3	2	1	0
Hardwired to 0x00h							

8.12.7. Header Type

HEADER_TYPE			Access = h			Regoffset = 0x0Eh	
7	6	5	4	3	2	1	0
Hardwired to 0x00h							

8.12.8. BIST

BIST			Access = h			Regoffset = 0x0Fh	
7	6	5	4	3	2	1	0
Hardwired to 0x00h							



8.12.9. Memory Base Address 0

MEMORY_BASE_ADDRESS																Access = h	Regoffset = 0X10h
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
MBA																	
Default value after reset = h																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MBA				Rsv											
Default value after reset = h															

Bit Number	Mnemonic	Description
Bits 31:12	MBA	Memory Base Address.
Bits 11:0	Rsv	Reserved, set to 0



8.12.10. I/O Base Address 0

I/O_BASE_ADDRESS Access = h Regoffset = 0X14h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I/OBA															
Default value after reset = h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I/OBA				Rsv											
Default value after reset = h															

Bit Number	Mnemonic	Description
Bits 31:12	I/OBA	I/O Base Address.
Bits 11:0	Rsv	Reserved, set to 0



8.12.11. Interrupt Line

This register is used in the manner defined in the PCI Local Bus Specification and PCI to PCI Bridge Specification

INTERRUPT_LINE				Access = h		Regoffset = 0x3Ch	
7	6	5	4	3	2	1	0
Writable by POST. If not implemented, hardwired to 0xFF							

8.12.12. Interrupt Pin

Read only register. The bit definition adheres to the PCI Local Bus Specification and PCI to PCI Bridge Specification.

INTERRUPT_PIN				Access = h		Regoffset = 0x3Dh	
7	6	5	4	3	2	1	0
Hardwired to 0x01 (INTA# interrupt used)							



9. ISA INTERFACE

9.1. INTRODUCTION

The ISA Interface provides access to the peripherals available in the STPC device and to Memory and external devices on the ISA bus.

Control of the ISA bus is by the North Bridge which acts as a bridge between the host CPU bus and the PCI bus. Reads and writes which are initiated by the CPU are subtractively decoded. Reads and writes that target North Bridge internal registers or main memory are routed to those targets, and all other reads and writes are sent to the PCI bus. The cycles for interrupt acknowledge, shutdown, stop grant and halt are also sent to the PCI bus.

The North Bridge also routes PCI reads and writes to cache, main memory and its internal registers. The South Bridge acts as a bridge between the PCI bus and the ISA bus. ISA bus cycles may be initiated by PCI bus cycles, or by an ISA bus card. Additionally, refresh cycles are run periodically by the ISA controller.

The South Bridge will claim all PCI cycles which were initiated outside the South Bridge and not claimed by any other PCI slave. Reads and writes to PCI configuration registers are routed appropriately by the South Bridge's PCI controller. All other PCI operations, including reads and writes to the South Bridge internal registers, are sent to the ISA controller. With the exception of writes to the keyboard controller under certain conditions, a read or write cycle sent to the ISA bus controller will create one or more ISA bus cycles.

Because of the speed difference between ISA bus and PCI bus, and the requirement that PCI cycles be less than a certain number of clocks, PCI cycles which go to the ISA bus will require retries on the PCI bus.

The cycles for interrupt acknowledge, shutdown, stop grant and halt are also sent to the ISA bus controller. These cycles do not create ISA bus cycles, but they use the same state machines for timing and arbitration as reads and writes. ISA bus cycles which are initiated by an ISA bus card are either DMA cycles, in which case the address is supplied by the DMA controller, or ISA bus master cycles, in which case the address is supplied by the card itself.

Every cycle initiated on the ISA bus is tried on the PCI bus. If the cycle is claimed by some PCI target, then data is read from or written to that target. If the PCI cycle is not claimed, and the cycle targets a South Bridge internal register, then that register is read from or written to. Otherwise, the target is expected to be on the ISA bus.

9.2. PCI / ISA CYCLES

9.2.1. PCI TO ISA READ AND WRITE

The PCI transfers data four bytes at a time, with byte enables for each byte. The South Bridge's PCI controller transfers these four bytes and four byte enables to the ISA controller. The ISA controller in turn runs zero to four ISA cycles. For 8-bit targets, the enabled bytes are read or written in order, least significant byte (lowest address) first.

For 16-bit targets, enabled bytes are again read or written in order, but a 16-bit transfer is used when an even byte is enabled and the following odd byte is also enabled.

Eight-bit ISA operations are by default four and a half ISACKLCK cycles, starting on a falling edge of ISACKLCK and ending on a rising edge. Sixteen-bit cycles are by default two and a half ISACKLCK cycles, also starting on a falling edge of ISACKLCK and ending on a rising clock. An additional clock cycle may be added by setting bit 5 in Index Register 50. Cycles can also be extended by pulling IOCHRDY low.



9.2.2. PCI TO INTERNAL REGISTER READ AND WRITE

All South Bridge internal registers are 8-bit. If an IO read or write targets an internal register, the target is assumed to be 8-bit wide (that is IOCS16# is ignored). Timing for reads and writes to internal registers is the same as 8-bit cycles on the ISA bus (see [Section 9.2.1.](#)).

If a write targets an internal register of the South Bridge, the data is written to the register and also to the ISA bus. If a read targets an internal register, the internal register is read, the South Bridge drives the ISA data bus with the contents of the register, and a ISA read cycle is done.

Registers that are called index registers in this document are indirectly addressed through a register at IO address 22h. There are two copies of this register, one on the North Bridge and one on the South Bridge.

Writes to IO address 22h go to both copies of the register. Reads from IO address 22h normally come from the North Bridge copy of the register, and do not generate a read cycle on the PCI bus. For test purposes, this behaviour can be changed by setting bit zero of index register 21h. In this case, a read from IO address 22h reads the South Bridge copy of the register, using a PCI read cycle.

After selecting an index register by writing to IO address 22h, that index register is read from or written to at IO address 23h. Some index registers are implemented in the North Bridge alone, some in the South Bridge alone, and some are duplicated and implemented in both. Whether an index register is implemented in the North Bridge, South Bridge or both is indicated in the description of that register in this document.

For index registers that are implemented in the North Bridge alone, writes to IO address 23h write to the register, and reads of IO address 23h read from the register, and no PCI cycles are generated.

For index registers that are implemented in the South Bridge alone, writes to IO address 23h write to the register, and reads of IO address 23h read from the register. In both cases, the data must go over the PCI bus.

For index registers that are implemented in both the North Bridge and the South Bridge, writes to IO address 23h write to both copies of the register, requiring a PCI write cycle. Reads to IO address 23h reads from the North Bridge copy of the register, and generate no PCI cycles. For test purposes, this behaviour can be changed by setting bit zero of index register 21h. In this case, the South Bridge copy of the register is read, using a PCI read cycle.

9.2.3. INTERRUPT ACKNOWLEDGE CYCLE

When an interrupt is requested, the interrupt controller in the South Bridge asserts the CPU's interrupt input. When the CPU services the interrupt, it must first get the interrupt vector from the interrupt controller. The interrupt vector is used to find the interrupt service routine. Also, since each interrupt request input of the interrupt controller has its own interrupt vector, the vector tells where the interrupt request came from.

To get the interrupt vector, the CPU generates two interrupt acknowledge cycles. Both of these cycles read data from the interrupt controller. The data returned by the first is ignored, while the data for the second contains the interrupt vector in bits 0-7. The North Bridge handles both of the cycles identically, converting them to PCI interrupt acknowledge cycles.

Outside of the interrupt controller, the South Bridge handles both cycles identically. The ISA controller converts the PCI cycles into interrupt acknowledge cycles for the interrupt controllers. The INTA# input of the interrupt controller is asserted for four and a half ISA bus clocks, starting on a falling edge of that clock, and during this time data is transferred from the interrupt controller to the ISA controller. This can be extended to five and a half clocks by setting bit 5 in Index Register 50h.

9.2.4. ISA TO PCI READ AND WRITE

ISA initiated cycles are converted to PCI cycles by the ISA controller. The South Bridge pulls IOCHRDY low to extend these cycles until the PCI cycle has completed.

9.2.5. ISA TO PCI BUFFERED READS

ISA reads of host memory can be buffered. This is disabled by default, and can be enabled by setting bit 6 in Index Register 50h. When this bit is set, ISA bus initiated reads of host memory addresses always get their data from a four byte buffer in the ISA controller which is filled on demand. This can reduce the amount of traffic for a block memory read by up to a factor of four.

The buffer is filled or refilled, under the conditions listed below, after the start of a ISA initiated read of a host memory address has been detected by the South Bridge. The South Bridge generates a PCI read of four bytes, with the low two bits of the address set to zero, and the rest of the address set to be the same as the address on the ISA bus address. The requested data will be driven by the South Bridge onto the ISA bus to finish the ISA read cycle.

The buffer will be refilled if the data requested by the current read is not in the buffer. Also, to avoid stale data, the buffer will be refilled for:

- The first host memory read after an ISA bus master gets ownership of the bus,
- The first host memory read after any ISA bus cycle which is not a host memory read,
- Any ISA read of a byte in the buffer which has already been read since the buffer was last filled.

If a host memory read can be fulfilled without refilling the buffer, no PCI cycle is generated.

9.2.6. ISA TO PCI POSTED WRITES

ISA writes to host memory can be posted. This is disabled by default, and can be enabled by setting bit 7 in Index Register 50h. When this bit is set, ISA bus initiated writes to host memory addresses go to a four byte buffer in the ISA controller. No PCI write is generated until the buffer is written to host memory.

The buffer is written to host memory when:

- The buffer gets full,
- or there is a host memory write to a location not in the buffer,
- or a host memory write would overwrite data already in the buffer,
- or there is an ISA cycle which is not a host memory write,
- or the current ISA master gives up ownership of the bus.

If writing the buffer to host memory is triggered by an ISA bus cycle, that cycle is held up by pulling IOCHRDY low until the buffer has been written to host memory.

Note that it is possible for the South Bridge to generate writes with discontinuous byte enables if posted writes are enabled.

9.2.7. ISA TO REGISTER READ AND WRITE

ISA initiated cycles which target South Bridge internal registers will first be tried on the PCI bus. If they are not claimed by a PCI target, then the register will be read or written. Reads and writes to IPC registers will cause the South Bridge to pull IOCHRDY low for at least the number of cycles programmed into Index Register 01h. Reads and writes to the South Bridge registers which are not IPC registers are normally

disabled. These can be enabled by setting bit 7 of Index Register 51h. Writes to these registers require a longer than standard recovery time of two ISACLK periods.

9.3. XBUS READ AND WRITE

The XBUS is an 8-bit subset of the ISA bus that connects low speed devices on the mother board to the CPU. In particular, the Real Time Clock (RTC), the Keyboard Controller, and the BIOS ROM will usually be connected via the XBUS. For the STPC, the XBUS shares address, data and command lines with the ISA bus. No buffers or transceivers are required to connect the XBUS to the ISA bus. The timing for XBUS cycles is the same as that for eight bit ISA cycles, see above.

9.3.1. REAL TIME CLOCK READ AND WRITE

The Real Time Clock (RTC) is connected to the XBUS. However the RTC is not connected to the command lines of the XBUS. Instead, four input pins of the RTC (CS#, AS, RW#, DS) are controlled directly by the STPC. The MOT pin of the RTC must be tied low. The registers in the RTC are accessed indirectly, by first writing the register number to IO port 70h, and then reading or writing the register at IO port 71h.

The RTC input CS# is connected to the logical OR of the outputs RMRTCCS# and ISAOE#. CS# is the chip select for the RTC, and it will be driven low (active) on any IO read or write to port 70h or port 71h, and also will be driven low by reads or writes to ROM address space.

The RTC input AS is directly connected to the RTCAS output. AS is the address strobe for the RTC, and it is asserted (high) during any IO write to port 70h.

The RTC input RW# is connected to the logical OR of the RTCRW# and ISAOE# outputs. RW# is write pulse for the RTC, and it will be asserted (low) during any IO write to port 71h.

The RTC input DS is connected to the logical OR of the South Bridge outputs RTCDS and ISAOE#. DS is the read pulse for the RTC, and it will be asserted (low) during any IO read of port 71h.

The RTC interrupt output IRQ# is directly connected to the IRQ8 input. There is an internal inverter between the pin IRQ8 and the interrupt controller to maintain compatibility with the PC-AT without requiring additional external glue logic.

9.3.2. BIOS ROM READ AND WRITE

The BIOS ROM is connected to the XBUS. The chip select for the ROM is connected to the logical OR of the RMRTCCS# and ISAOE# outputs.

9.3.3. CPU RESET AND GATE A20

Before the 286 CPU, memory space was limited to 1MB. Some software applications used this characteristic to access data in segment 0 by generating an address above the 1MB. To stay compatible with these applications, when the 286 appeared, the PC motherboards included, via the keyboard controller, a mechanism to enable or disable this pre-286 compatibility. This is done by the 'Gate A20' mechanism. When enabled, the CPU A20 address line is propagated to the memory bus. When disabled, the memory bus A20 line is forced to 0 (8086 compatible).

To be able to reset the CPU, the keyboard controller also includes a pin which is connected to the CPU reset pin (only the CPU is reset, not the chipset or external components).

The STPC doesn't provide external pins to be able to control the gate A20 and CPU reset. These features are controlled internally by the keyboard emulation. Thus, the STPC checks the commands and data sent to the keyboard controller, and when it recognizes the related commands, applies them internally. The keyboard emulation must be on (register STPC_MISC0/bit 3 = 0), else you won't be able to reset the CPU and the A20 line will always be masked. This is done in the default configuration.

Notes: Only the P2 write command (D1h) and the reset pulse command (FEh) are emulated. In particular, the P2 read command (D0h) is not emulated, so the return value is the keyboard controller P2 state.

9.3.3.1. Reset Method

To disable Gate A20 (forcing address bit 20 to low), write D1h to the I/O Port 64h then write xxxx xx0xb to I/O Port 60h.

To enable Gate A20 (forcing address bit 20h to high

The Reset, also known as warm reset, is generated by writing data FEh to I/O port 64h or by writing data DEh to I/O port 64h then writing data xxxxxxx0 binary (bit 0 = '0') to I/O port 60h.

Fast host CPU reset only is generated by two methods:

- 1) Whenever the STPC detects a write to Port 64h with data FEh.
- 2) Whenever the STPC detects a write to Port 60h following a D1h data write to Port 64h, bit 0 of the data byte being written at Port 60h is '0'.

The CPU reset is at least 16 host clocks. The write cycle is not forwarded to the keyboard controller.

9.4. ISA STANDARD REGISTERS

The ISA standard registers correspond to the registers in the peripheral components integrated in the STPC as well as the miscellaneous ports implemented on a ISA motherboard. These registers reside in IO space.

The functions controlled by the ISA registers include the DMA and interrupt control, BIOS and keyboard interface.

9.4.1. DMA 1 CONTROLLER REGISTERS

DMA 1 controls 8-bit DMA transfers.

There are 16 DMA 1 registers. They are as shown in [Table 9-1](#).

Table 9-1. DMA1 Registers

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 000x 0000	xxxx xxxx	DMA 1 Channel 0 Base and Current Address	DMA1_CBA0
XXXX XX00 000x 0001	xxxx xxxx	DMA 1 Channel 0 Base and Current Count	DMA1_CBC0
XXXX XX00 000x 0010	xxxx xxxx	DMA 1 Channel 1 Base and Current Address	DMA1_CBA1
XXXX XX00 000x 0011	xxxx xxxx	DMA 1 Channel 1 Base and Current Count	DMA1_CBC1
XXXX XX00 000x 0100	xxxx xxxx	DMA 1 Channel 2 Base and Current Address	DMA1_CBA2
XXXX XX00 000x 0101	xxxx xxxx	DMA 1 Channel 2 Base and Current Count	DMA1_CBC2
XXXX XX00 000x 0110	xxxx xxxx	DMA 1 Channel 3 Base and Current Address	DMA1_CBA3
XXXX XX00 000x 0111	xxxx xxxx	DMA 1 Channel 3 Base and Current Count	DMA1_CBC3
XXXX XX00 000x 1000	xxxx 0000	DMA 1 Read Status/Write Command register	DMA1_RSWC
XXXX XX00 000x 1001	1111 xxxx	DMA 1 Request register	DMA1_RR
XXXX XX00 000x 1010	0000 0000	DMA 1 Read Command/Write Single Mask register	DMA1_RCWSM
XXXX XX00 000x 1011	0000 0000	DMA 1 Mode register	DMA1_Mode
XXXX XX00 000x 1100	1111 1111	DMA 1 Set/Clear Byte pointer flip-flop	DMA1_SCBPFF
XXXX XX00 000x 1101	0000 0000	DMA 1 Read Temp register/Master Clear	DMA1_RTMC
XXXX XX00 000x 1110	1111 1111	DMA 1 Clear Mask/Clear all request	DMA1_CMCAR
XXXX XX00 000x 1111	1111 1111	DMA 1 Read/Write all Mask register bits	DMA1_RWMB

Note that the not all bits of the address are used.

Programming notes:

Channel 0 corresponds to the internal DRQ0B, channel 1 to DRQ1B, channel 2 to DRQ2B, and channel 3 corresponds to the internal DRQ3B.



9.4.2. INTERRUPT CONTROLLER 1 REGISTERS

There are two interrupt controller 1 registers. They are as shown in [Table 9-2](#).

Interrupt controller 1 is the master interrupt controller.

Table 9-2. Interrupt Controller 1 Registers

IO address bits 15-0	Reset Value	Register Name	Mnemonic
0000 0000 0010 0000	0000 0000	Interrupt Controller 1 register	IC_1
0000 0000 0010 0001	1111 1111	Interrupt Controller 1 Mask register	IC_1MR

Note that not all bits of the address are used.

Programming notes:

Interrupt controller 1 input IR0 is connected IRQ0, IR1 to IRQ1, IR2 to interrupt out from interrupt controller 2, IR3 to IRQ3, IR4 to IRQ4, IR5 to IRQ5, IR6 to IRQ6, and IR7 to IRQ7.



9.4.3. INTERVAL TIMER REGISTERS

The Interval Timer comprises three independent counters. Counter 0 is used to generate timer interrupts, counter 1 is used to generate ISA bus refresh, and counter 2 is used to create the speaker tone.

There are 4 Interval Timer registers. They are as shown in [Table 9-3](#).

Table 9-3. Interval Timer Registers

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 010x xx00	xxxx xxxx	Interval Timer Register Counter 0 Count	IT_0
XXXX XX00 010x xx01	xxxx xxxx	Interval Timer Register Counter 1 Count	IT_1
XXXX XX00 010x xx10	xxxx xxxx	Interval Timer Register Counter 2 Count	IT_2
XXXX XX00 010x xx11	1111 1111	Command Mode register	IT_3

Note that not all bits of the address are decoded.

Programming notes:

All three counters are clocked by 1.193 MHz nominal frequency (OSC/12). Counter 0 and counter 1 gates are always on, counter 2 gate is controlled by writing to Port B (see [Section 9.4.4](#)).



9.4.4. PORT B REGISTER

This is the ISA compatible 8-bit Port B register located at xxxx xxxx 0110 xxx1 IO address (bits 15-0). It has the following meaning:

Port_B			Access = 0061h				Regoffset =	
7	6	5	4	3	2	1	0	
PE	IOCHK	T/C 2S	ISA RC	ISA IOCHK	PCE	SE	T/C 2 G	
Default value after reset = 00h								

Bit Number	Mnemonic	Description
Bit 7	PE	Parity Error. This bit is set to a '1' whenever a parity error is detected during system memory read operation. Once set, this bit can be cleared by setting bit 2 of this register to a '1'. Bit 2 should be reset to a '0' to enable recording the next parity error. The parity error generates NMI to the host CPU if NMI is enabled. This bit is read-only.
Bit 6	IOCHK	ISA IOCHK# Enable. This bit is set to a '1' when IOCHK# signal of the ISA bus is asserted. Once set, this bit is cleared by setting bit 3 of this register to a '1'. Bit 3 should be reset to a '0' to enable recording the next IOCHK#. IOCHK# generates NMI to the host CPU if NMI is enabled. This bit is read only.
Bit 5	T/C 2S	ISA T/C 2 State. This bit reflects the output of Timer/Counter 2 without any synchronization. This bit is read only.
Bit 4	ISA RC	ISA Refresh Check. This bit toggles on every rising edge of the REFRESH# signal of the ISA bus. This bit is read only.
Bit 3	ISA IOCHK	ISA IOCHK# Enable. This bit is connected to the asynchronous clear input of the flipflop which records the IOCHK#. It must be set to a '1' to clear the flipflop and then set to a '0' to enable further IOCHK# assertions. This bit is read/write and cleared to a '0' by ISA reset.
Bit 2	PCE	Parity Check Enable. This bit is connected to the asynchronous clear input of the flipflop which records the parity error. It must be set to a '1' to clear the flipflop and then set to a '0' to enable further parity errors. This bit is read/write and cleared to a '0' by ISA reset.
Bit 1	SE	ISA Speaker Enable. This bit is ANDed with the Interval Timer counter 2 OUT signal to drive the Speaker output signal. This bit is read/write and cleared to a '0' by ISA reset.
Bit 0	T/C 2G	T/C 2 Gate. This bit is connected to the gate input of the Interval Timer counter 2. This bit is read/write and cleared to a '0' by ISA reset.

9.4.5. PORT 70H REGISTER

This 8-bit write-only register contains the NMI enable bit and is located at xxxx xxxx 0111 0xx1 IO address.

Port_70			Access = 0070h			Regoffset =	
7	6	5	4	3	2	1	0
NMI E	Rsv						
Default value after reset = 80h							

Bit Number	Mnemonic	Description
Bit 7	NMI E	NMI Enable. NMI is asserted on encountering IOCHK# on the ISA bus (Port_B) or SERR# on the PCI bus if this bit is set to a '0'. Setting this bit to a '1' disables NMI generation.
Bit 6-0	Rsv	Reserved. must be written to '0's. Read back is undefined.

Programming notes:

Writing to this address also sets the address register in the Real Time Clock (RTC, not part of the STPC, normally connected via the ISA interface).



9.4.6. INTERRUPT CONTROLLER 2 REGISTERS

Interrupt controller 2 is the slave interrupt controller.

Interrupt controller 2 occupies two register locations. They are as shown in [Table 9-4](#).

Table 9-4. Interrupt Controller 2 Registers

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 101x xxx0	0000 0000	Interrupt Controller 2 register	IC_2R
XXXX XX00 101x xxx1	1111 1111	Interrupt Controller 2 Mask register	IC_2M

Note that not all address bits are decoded.

Programming notes:

Interrupt controller 2 input IR1 is connected to IRQ9, IR2 to IRQ10, IR3 to IRQ11, IR4 to IRQ12, IR6 to IRQ14, IR7 to IRQ15. IR0 driven by IRQ8 inverted. IR5 is driven by an internally generated floating point error interrupt request.



9.4.7. DMA CONTROLLER 2 REGISTERS

There are 16 DMA 2 registers. They are as shown in [Table 9-5](#).

Table 9-5. DMA Controller 2 Registers

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 1100 000x	xxxx xxxx	DMA 2 Channel 0 Base and Current Address	DMA2_CBA0
XXXX XX00 1100 001x	xxxx xxxx	DMA 2 Channel 0 Base and Current Count	DMA2_CBC0
XXXX XX00 1100 010x	xxxx xxxx	DMA 2 Channel 1 Base and Current Address	DMA2_CBA1
XXXX XX00 1100 011x	xxxx xxxx	DMA 2 Channel 1 Base and Current	DMA2_CBC1
XXXX XX00 1100 100x	xxxx xxxx	DMA 2 Channel 2 Base and Current Address	DMA2_CBA2
XXXX XX00 1100 101x	xxxx xxxx	DMA 2 Channel 2 Base and Current	DMA2_CBC2
XXXX XX00 1100 110x	xxxx xxxx	DMA 2 Channel 3 Base and Current Address	DMA2_CBA3
XXXX XX00 1100 111x	xxxx xxxx	DMA 2 Channel 3 Base and Current Count	DMA2_CBC3
XXXX XX00 1101 000x	1111 xxxx	DMA 2 Read Status/Write Command register	DMA2_RSWC
XXXX XX00 1101 001x	0000 0000	DMA 2 Request register	DMA2_RR
XXXX XX00 1101 010x	0000 0000	DMA 2 Read Command/Write Single Mask register	DMA2_RCWSM
XXXX XX00 1101 011x	0000 0000	DMA 2 Mode register	DMA2_Mode
XXXX XX00 1101 100x	1111 1111	DMA 2 Set/Clear Byte pointer flip-flop	DMA2_SCBPFF
XXXX XX00 1101 101x	0000 0000	DMA 2 Read Temporary/Master Clear	DMA2_RTMC
XXXX XX00 1101 110x	1111 1111	DMA 2 Clear Mask/Clear all requests register	DMA2_CMCAR
XXXX XX00 1101 111x	1111 1111	DMA 2 Read/Write all Mask register bits	DMA2_RWMRB

Note that the not all bits of the address are used.

9.4.8. DMA PAGE REGISTERS

The DMA Page registers defines address bits [16-23] for DMA transfers controlled by DMA 1 or DMA 2. Bits [0-15] are generated by the DMA controller, bits [16-23] come from the appropriate page register, and bits 31-24 are all zeroes.

There are 16 DMA page registers. They are as shown in [Table 9-6](#).

Table 9-6. DMA Page Registers

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 0000	xxxx xxxx	DMA Page Register Port 80h (Reserved)	Port_80
XXXX XX00 0001	xxxx xxxx	DMA Page Register Channel 2	DMA_PRC2
XXXX XX00 0010	xxxx xxxx	DMA Page Register Channel 3	DMA_PRC3
XXXX XX00 0011	xxxx xxxx	DMA Page Register Channel 1	DMA_PRC1
XXXX XX00 0100	xxxx xxxx	DMA Page Register Port 84h (Reserved)	Port_84
XXXX XX00 0101	xxxx xxxx	DMA Page Register Port 85h (Reserved)	Port_85
XXXX XX00 0110	xxxx xxxx	DMA Page Register Port 86h (Reserved)	Port_86
XXXX XX00 0111	xxxx xxxx	DMA Page Register Channel 0	DMA_PRC0
XXXX XX00 1000	xxxx xxxx	DMA Page Register Port 87h (Reserved)	Port_87
XXXX XX00 1001	xxxx xxxx	DMA Page Register Channel 6	DMA_PRC6
XXXX XX00 1010	xxxx xxxx	DMA Page Register Channel 7	DMA_PRC7
XXXX XX00 1011	xxxx xxxx	DMA Page Register Channel 5	DMA_PRC5
XXXX XX00 1000 1100	xxxx xxxx	DMA Page Register Port 8Bh (Reserved)	Port_8B
XXXX XX00 1000 1101	xxxx xxxx	DMA Page Register Port 8Ch (Reserved)	Port_8C
XXXX XX00 1000 1110	xxxx xxxx	DMA Page Register Port 8Dh (Reserved)	Port_8D
XXXX XX00 1000 1111	xxxx xxxx	DMA Page Register Port 8Eh (Reserved)	Port_8E

9.5. ISA CONFIGURATION REGISTERS

These registers are addressed through the Address Configuration Index (CI) and Data registers.

9.5.1. MISCELLANEOUS CONTROL REGISTER 0

Misc_Cont0			Access = 0022h/0023h			Regoffset = 050h	
7	6	5	4	3	2	1	0
ISA WPE	ISA RBE	ISA WIC	ISA CFS	KRE	CPU D		
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	ISA WPE	ISA Write Post Enable. If '1', posted writes to host memory by ISA DMA or ISA bus master are enabled.
Bit 6	ISA RBE	ISA Read Buffer Enable. If '1', buffered reads of host memory by ISA DMA or ISA bus master are enabled.
Bit 5	ISA WIC	ISA Wait Insert Control. This bit controls if extra wait state is inserted for slower ISA devices. 0: No extra wait state for ISA cycle 1: One extra wait state for ISA cycle
Bit 4	ISA CFS	ISA Clock Frequency Select. This bit selects the ISA clock frequency. 0: ISA clock is 14.31818 MHz / 2 1: ISA clock is PCICLK /4
Bit 3	KRE	Keyboard Reset Enable. This bit if set to a '1', keyboard emulation fast gate A20 and fast reset are disabled. The source of warm reset indication is from the external keyboard controller and the CPU core will use the gate A20 indication from keyboard controller for its internal A20M# input.
Bits 2-0	CPU D	CPU Deturbo. These three bits define the ratio CPU is held. (see Table 9-7).

Table 9-7. CPU Deturbo

Bit 2	Bit 1	Bit 0	CPU Deturbo
0	0	0	deturbo is disabled.
0	0	1	CPU is held 1/2 of the time.
0	1	0	CPU is held 2/3 of the time.
0	1	1	CPU is held 3/4 of the time.
1	0	0	CPU is held 4/5 of the time.
1	0	1	CPU is held 5/6 of the time.
1	1	0	CPU is held 6/7 of the time.
1	1	1	CPU is held 7/8 of the time.



9.5.2. MISCELLANEOUS CONTROL REGISTER 1

Misc_Cont1

Access = 0022h/0023h

Regoffset = 051h

7	6	5	4	3	2	1	0
IPC W	CLK 24	HCLK D	Rsv	ROM	S E S	S D S	S C S
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	IPC W	IPC Write control. This bit controls the ISA master writes to the IPC register 0: ISA master writes to IPC register disabled 1: ISA master writes to IPC register enabled
Bit 6	CLK 24	CLK24 Disable. This bit controls the output of CLK24. 0: CLK24 generated normally 1: Clock synthesiser for CLK24 is disable (CLK24 will not toggle)
Bit 5	HCLK D	HCLK Disable. This bit controls the generation of HCLK. 0: HCLK generated normally 1: Clock synthesiser for HCLK is disabled (HCLK will not toggle)
Bit 4	Rsv	Reserved.
Bit 3	ROM	ROM Write Protect Enable. This bit, if set to a '1', disables write cycles to ROM BIOS on extended bus. If set to '0', write to extended bus ROM BIOS is allowed. Note: This bit can not disable the write to shadowed BIOS in DRAM since after shadow is enabled, all writes to BIOS should be forwarded to extended bus.
Bit 2	S E S	Segment E Share. This bit controls if E0000h-EFFFFh segment shares the FLASH memory with F0000h-FFFFFh segment. 0: Sharing disabled 1: Sharing enabled
Bit 1	S D S	Segment D Share. This bit controls if D0000h-DFFFFh segment shares the FLASH memory with F0000h-FFFFFh segment. 0: Sharing disabled 1: Sharing enabled
Bit 0	S C S	Segment C Share. This bit controls if C0000h-CFFFFh segment shares the FLASH memory with F0000h-FFFFFh segment. 0: Sharing disabled 1: Sharing enabled

9.5.3. PIRQ ROUTING CONTROL REGISTER 0

This 8-bit register controls the routing of PCI Interrupt A# to one of the interrupt inputs of the 8259 as follows. It applies to interrupts A# to D#:

PIRQ Routing	Regoffset
PIRQ A	52h
PIRQ B	53h
PIRQ C	54h
PIRQ D	55h

PAR_Cont0 Access = 0022h/0023h Regoffset = [Table 8-1](#).

7	6	5	4	3	2	1	0
RE	Rsv			RC A			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	RE	Routing Enable. If set to a '1', this bit enables the routing of PCI interrupt, otherwise the PCI interrupt A# is unconnected.
Bits 6-4	Rsv	Reserved. Writes have no affect. Reads return undefined value.
Bits 3-0	RC A	Routing Control. These bits route the PCI interrupt A# (see Table 9-8)

Table 9-8. Routing Control Encoding

Bit 3	Bit 2	Bit 1	Bit 0	Interrupt A# Route	Note
0	0	0	1	Reserved	1
0	0	0	1	Reserved	1
0	0	1	0	Reserved	1
0	0	1	1	IRQ3	
0	1	0	0	IRQ4	
0	1	0	1	IRQ5	
0	1	1	0	IRQ6	
0	1	1	1	IRQ7	
1	0	0	0	Reserved.	1
1	0	0	1	IRQ9	
1	0	1	0	IRQ10	
1	0	1	1	IRQ11	
1	1	0	0	IRQ12	
1	1	0	1	Reserved.	1
1	1	1	0	IRQ14	
1	1	1	1	IRQ15	

Note 1: Interrupt can not be routed to this level.



9.5.4. INTERRUPT LEVEL CONTROL REGISTER 0

This 8-bit register allows interrupt requests to the lower 8259 to be either level or edge sensitive on an interrupt by interrupt basis overriding the global edge/level control bit of the 8259.

IRQ_Lev_C_0			Access = 0022h/0023h			Regoffset = 056h	
7	6	5	4	3	2	1	0
IRQ C					Rsv		
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-3	IRQ C	IRQ Control IRQ[7-3]. If set to a '1', the corresponding interrupt request is treated as a level input, otherwise it is treated as the edge sensitive input (compatible to ISA).
Bits 2-0	Rsv	Reserved. Writes have no affect. Reads return undefined value.



9.5.5. INTERRUPT LEVEL CONTROL REGISTER 1

This register allows interrupt requests to the upper 8259 to be either level or edge sensitive on an interrupt by interrupt basis overriding the global edge/level control bit of the 8259. This register has the following definition.

IRQ_Lev_C_1		Access = 0022h/0023h				Regoffset = 057h	
7	6	5	4	3	2	1	0
IRQ C		Rsv	IRQ C				IPC
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	IRQ C	IRQ Control IRQ[15-14]. If set to a '1', the corresponding interrupt request is treated as a level input, otherwise it is treated as the edge sensitive input (compatible with ISA).
Bit 5	Rsv	Reserved. Writes have no affect and the reads return undefined value.
Bits 4-1	IRQ C	IRQ Control IRQ[12-9]. If set to a '1', the corresponding interrupt request is treated as a level input, otherwise it is treated as the edge sensitive input (compatible to ISA).
Bit 0	IPC	This bit controls the ISA refresh cycle.Setting to 0 disables ISA refresh and setting to 1 enables ISA refresh. By setting this bit to 1 enables the toggling the ISA Port B refresh bit (see Section 9.4.4.).



9.5.6. IPC CONFIGURATION REGISTER

This 8-bit register controls the timing of the DMA controllers, and also the number of wait states for writes to registers in the IPC.

IPC_Conf		Access = 0022h/0023h				Regoffset = 001h	
7	6	5	4	3	2	1	0
IPC WS		DMA		DMA		DMA M	DMA C
Default value after reset = C0h							

Bit Number	Mnemonic	Description
Bits 7-6	IPC WS	IPC Wait States. These bits specify the number of ISACLK wait states for read or write to IPC register1 (see Table 9-9).
Bits 5-4	DMA	DMA 16-Bit Wait States. These bits specify the number of wait states in 16-bit DMA cycles (see Table 9-10).
Bits 3-2	DMA	DMA 8-Bit Wait States. These bits specify the number of wait states in 8 bit DMA cycle (see Table 9-11).
Bit 1	DMA M	DMA MEMR# Timing. If this bit is set to '1' the DMA controllers will assert MEMR# at the the same time as IOW#. If set to '0' (default), MEMR# will be asserted one clock after IOW#.
Bit 0	DMA C	DMA Clock Select. If this bit is set to '0' (default), the DMA controller clock will be ISACLK divided by two, otherwise the DMA controller clock will be ISACLK.

Table 9-9. IPC Wait States

Bit 7	Bit 6	IPC Wait States
0	0	1
0	1	2
1	0	3
1	1	4 (Default)

Table 9-10. DMA 16-bit Wait States

Bit 5	Bit 4	DMA 16-bit Wait States
0	0	1 (Default)
0	1	2
1	0	3
1	1	4



Table 9-11. DMA 8-bit Wait States

Bit 3	Bit 2	DMA 8-bit Wait States
0	0	1 (Default)
0	1	2
1	0	3
1	1	4

Programming notes:

To read or write to this register, write 01 to index register 22h, and then read or write from data register 23h.

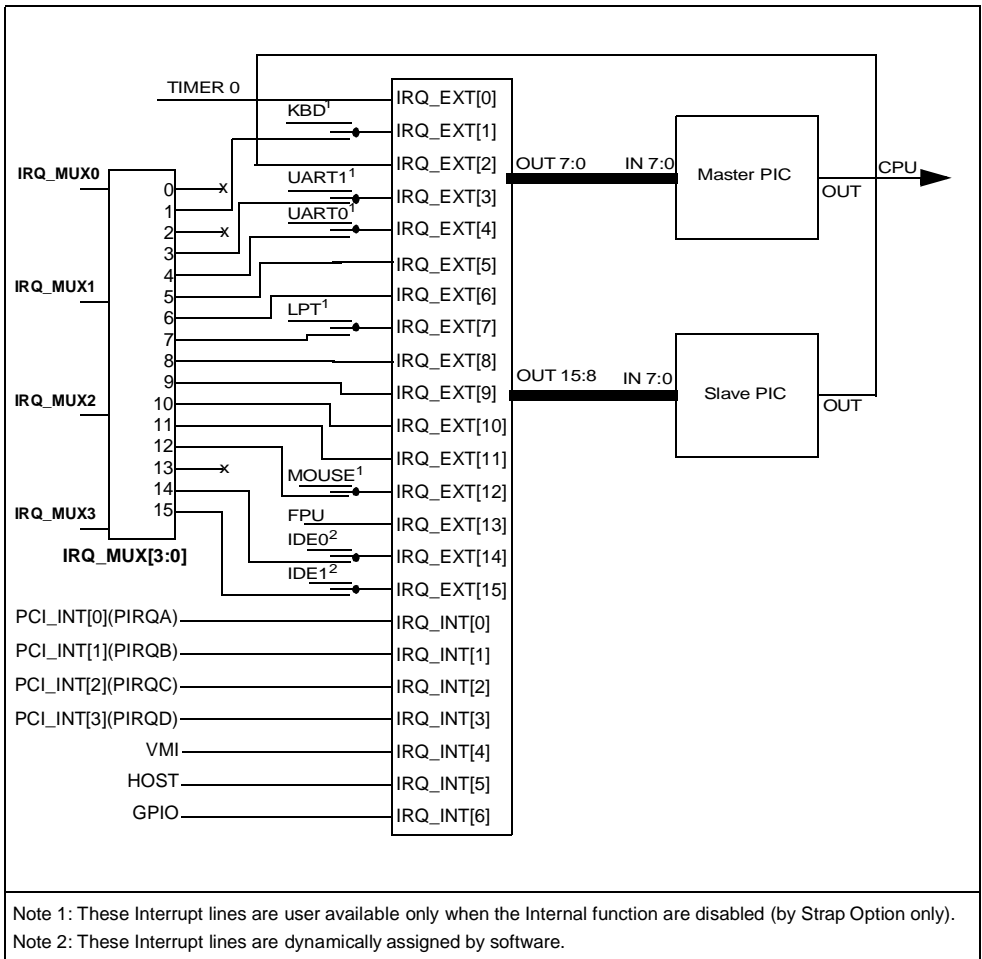


9.6. INTERRUPT ROUTER

A general purpose crossbar is implemented for the routing of internal and external IRQ interrupts to any of the 16 inputs of two 8259 Programmable Interrupt Controller (PIC) devices. This is represented in. The main interrupt features are listed below:

- No hardwired interrupt requirement
- Complete IRQ programming flexibility
- Interfaces with up to 11 internal (IRQ_INT[10:0]) and 16 external (IRQ_EXT[15-0]) interrupt requesting devices

Figure 9-1. Interrupt Router Schematic Layout



9.6.1. INTERRUPT CONTROLLER REGISTER SUMMARY

The Internal and External Interrupt Registers are set out as per the table below. The interrupts are layed out on their priority basis. If there are two or more input lines routed to the same output, only the highest priority input will be taken into account, the lower inputs will be ignored. In [Table 9-12](#), the highest priority is placed first the lowest priority is last.

Table 9-12. Interrupt Routing Control Registers

Register Offset/ Access	Default Value after Reset	Interrupt	Register Name	Mnemonic
Offset 052h ³	0x00h	IRQ_INT[0]	PIRQA Routing Control Register	<i>PAR_cont0</i>
Offset 053h ³	0x00h	IRQ_INT[1]	PIRQB Routing Control Register	<i>PBR_cont0</i>
Offset 054h ³	0x00h	IRQ_INT[2]	PIRQC Routing Control Register	<i>PCR_cont0</i>
Offset 055h ³	0x00h	IRQ_INT[3]	PIRQD Routing Control Register	<i>PDR_cont0</i>
Offset 058h ³	0x00h	IRQ_INT[4]	I ² C IRQ Routing Control Register	<i>VIR_cont</i>
0x410h	0x00h	IRQ_INT[5]	Not Implemented	<i>IRQ_INT5</i>
0x411h	0x00h	IRQ_INT[6]	GPIO IRQ Routing Control Register	<i>IRQ_INT6</i>
0x412h	0x00h	IRQ_INT[7]	Not Implemented	<i>IRQ_INT7</i>
0x413h	0x00h	IRQ_INT[8]	Not Implemented	<i>IRQ_INT8</i>
0x414h	0x00h	IRQ_INT[9]	Not Implemented	<i>IRQ_INT9</i>
0x415h	0x00h	IRQ_INT[10]	Not Implemented	<i>IRQ_INT10</i>
0x400h	0x80h	IRQ_EXT[0]	Timer 0 IRQ Routing Control Register	<i>IRQ_EXT0</i>
0x401h	0x81h	IRQ_EXT[1]	Keyboard or IRQ External Routing Control Register ^{1,2}	<i>IRQ_EXT1</i>
0x402h	0x82h	IRQ_EXT[2]	Slave PIC IRQ Routing Control Register	<i>IRQ_EXT2</i>
0x403h	0x83h	IRQ_EXT[3]	UART1 or IRQ External Routing Control Register ³	<i>IRQ_EXT3</i>
0x404h	0x84h	IRQ_EXT[4]	UART0 or IRQ External Routing Control Register ⁴	<i>IRQ_EXT4</i>
0x405h	0x85h	IRQ_EXT[5]	IRQ External Routing Control Register 5	<i>IRQ_EXT5</i>
0x406h	0x86h	IRQ_EXT[6]	IRQ External Routing Control Register 6	<i>IRQ_EXT6</i>
0x407h	0x87h	IRQ_EXT[7]	LPT or IRQ External Routing Control Register ^{7,2}	<i>IRQ_EXT7</i>
0x408h	0x88h	IRQ_EXT[8]	IRQ External Routing Control Register 8	<i>IRQ_EXT8</i>
0x409h	0x89h	IRQ_EXT[9]	IRQ External Routing Control Register 9	<i>IRQ_EXT9</i>
0x40Ah	0x8Ah	IRQ_EXT[10]	IRQ External Routing Control Register 10	<i>IRQ_EXT10</i>
0x40Bh	0x8Bh	IRQ_EXT[11]	IRQ External Routing Control Register 11	<i>IRQ_EXT11</i>
0x40Ch	0x8Ch	IRQ_EXT[12]	Mouse or External IRQ Routing Control Register ¹²	<i>IRQ_EXT12</i>
0x40Dh	0x8Dh	IRQ_EXT[13]	FPU IRQ Routing Control Register	<i>IRQ_EXT13</i>
0x40Eh	0x8Eh	IRQ_EXT[14]	PCI IDE0 or External IRQ Routing Control Register ¹	<i>IRQ_EXT14</i>
0x40Fh	0x8Fh	IRQ_EXT[15]	PCI IDE1 or External IRQ Routing Control Register ¹	<i>IRQ_EXT15</i>
The pre-assigned interrupts follow the PC standard. Seein the Datasheet for routing details. The interrupts can be re-used internally provided they are not required for specific PC operations				
Note 1; These interrupts are not available when the IDE is in legacy mode				
Note 2; These interrupts are available for the ISA Bus only if the corresponding integrated function is deactivated by Strap Option (seeSTRAP OPTIONS chapter of the Datasheet)				
Note 3; These registers are accessible through chip set access 22h/23h.				

The interrupt router is connected between the interrupt-requesting devices and the two 8259 PIC devices. It requires 27 registers, one for each input. Five of these registers, Internal [0] to Internal [4], are implemented as PCI_IRQA, PCI_IRQB, PCI_IRQC, PCI_IRQD and VMI_IRQ respectively (highest priority interrupts listed in [Table 9-12](#)), in I/O space 22h/23h. The remaining 22 registers are implemented in I/O space from 400h to 415h. Details of all 27 registers are given in the following sections.

9.6.2. INTERRUPT ROUTING CONTROL REGISTERS

These eleven 8-bit registers control the routing of internal interrupts IRQ_INT[10:0] and the sixteen external interrupts IRQ_EXT[15:0] to the interrupt inputs of the 8259 as in [Table 9-12](#). and [Figure 9-1](#).

Routing Control Register

Access = see [Table 9-12](#)

Regoffset see [Table 9-12](#)

7	6	5	4	3	2	1	0
RE	Rsv			RC			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	RE	Routing Enable. If set to a '1', this bit enables the routing of the internal interrupt, otherwise the internal interrupt is unconnected.
Bits 6-4	Rsv	Reserved. Writes have no affect. Reads return undefined value.
Bits 3-0	RC	Routing Control. These bits route the internal interrupt (see Table 9-8)

Table 9-13. Interrupt Route

Bit 3	Bit 2	Bit 1	Bit 0	Interrupt Route
0	0	0	0	IRQ0
0	0	0	1	IRQ1
0	0	1	0	IRQ2
0	0	1	1	IRQ3
0	1	0	0	IRQ4
0	1	0	1	IRQ5
0	1	1	0	IRQ6
0	1	1	1	IRQ7
1	0	0	0	IRQ8
1	0	0	1	IRQ9
1	0	1	0	IRQ10
1	0	1	1	IRQ11
1	1	0	0	IRQ12
1	1	0	1	IRQ13
1	1	1	0	IRQ14
1	1	1	1	IRQ15

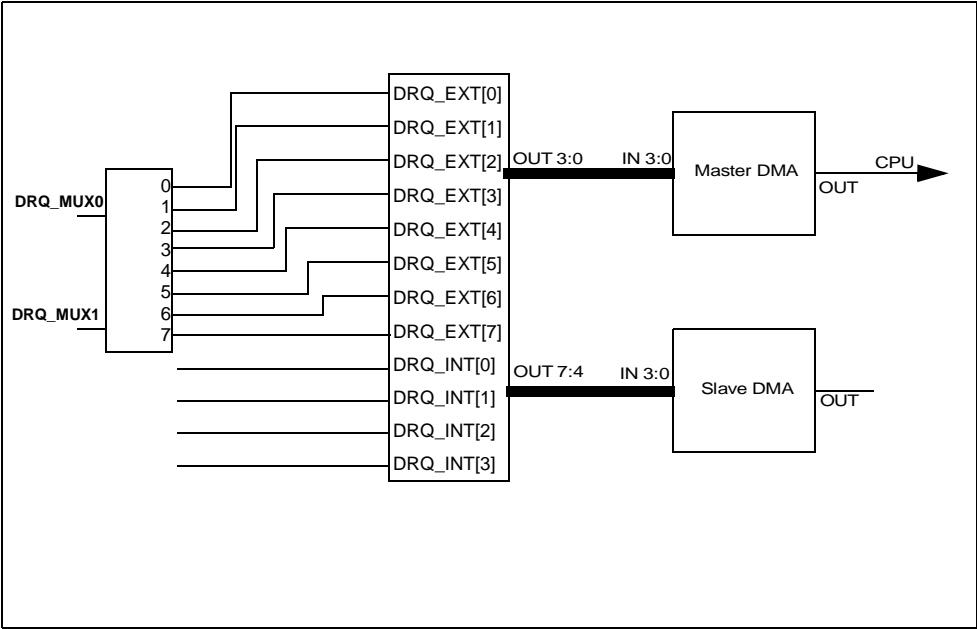


9.7. DRQ ROUTER

A general purpose crossbar is implemented for the routing of internal and external DRQs (DMA Requests) to any of the eight inputs of two 8237 DMA Controller devices. The main DRQ facility features are listed below:

- No hardwired DRQ requirement.
- Complete DRQ programming flexibility.
- Interfaces with internal and external DMA requesting devices.
- Number of DRQ lines expanded (in terms of the number of devices that can issue a DRQ) to eight external and four internal.

Figure 9-2. Interrupt Router Schematic Layout



9.7.1. DRQ INTERRUPT ROUTER SUMMARY

The internal and external DRQs are set out as per [Table 9-14](#). below. The DRQs are layed out on their priority basis. If there are two or more input lines routed to the same output, only the highest priority input will be taken into account, the lower inputs will be ignored. The highest priority is placed first and the lowest is last.

Table 9-14. DRQ Routing Control Registers

Access	Default Value after Reset	DRQ	Register Name	Mnemonic
0x428h	00h	DRQ_INT[0]	DRQ Internal Routing Control Register 0	IDRQ_cont0
0x429h	00h	DRQ_INT[1]	DRQ Internal Routing Control Register 1	IDRQ_cont1
0x42Ah	00h	DRQ_INT[2]	DRQ Internal Routing Control Register 2	IDRQ_cont2
0x42Bh	00h	DRQ_INT[3]	DRQ Internal Routing Control Register 3	IDRQ_cont3
0x420h	80h	DRQ_EXT[0]	DRQ External Routing Control Register 0	EDRQ_cont0
0x421h	81h	DRQ_EXT[1]	DRQ External Routing Control Register 1	EDRQ_cont1
0x422h	82h	DRQ_EXT[2]	DRQ External Routing Control Register 2	EDRQ_cont2
0x423h	83h	DRQ_EXT[3]	DRQ External Routing Control Register 3	EDRQ_cont3
0x424h	84h	DRQ_EXT[4]	DRQ External Routing Control Register 4	EDRQ_cont4
0x425h	85h	DRQ_EXT[5]	DRQ External Routing Control Register 5	EDRQ_cont5
0x426h	86h	DRQ_EXT[6]	DRQ External Routing Control Register 6	EDRQ_cont6
0x427h	87h	DRQ_EXT[7]	DRQ External Routing Control Register 7	EDRQ_cont7

The DRQ router is connected between the DRQ-requesting devices and the two 8237 DMA Controller devices. It requires 12 registers, one for each DRQ input (eight external, four internal). These registers are implemented from 420h to 42Ch. Details of all 12 registers are given in [Table 9-14](#).

9.7.2. DRQ ROUTING CONTROL REGISTERS

These four 8-bit registers control the routing of Internal DRQ0 to DRQ3 (DRQ_INT[0] to DRQ_INT[3]) and External Interrupts DRQ0 to DRQ7 (DRQ_EXT[0] to DRQ_EXT[7]) to one of the DRQ inputs of the 8237 as follows:

IDRQ_cont0

Access: see [Table 9-14](#)

7	6	5	4	3	2	1	0
RE	Rsv				RC		
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	RE	Routing Enable. If set to a '1', this bit enables the routing of the Internal DRQ, otherwise the DRQ is unconnected.
Bits 6-3	Rsv	Reserved. Writes have no affect. Reads return undefined value.
Bits 2-0	RC	Routing Control. These bits route the IDRQ (see Table 9-15)

Table 9-15. IDRQ Route

Bit 2	Bit 1	Bit 0	Internal DRQ Route
0	0	0	DRQ0
0	0	1	DRQ1
0	1	0	DRQ2
0	1	1	DRQ3
1	0	0	DRQ4
1	0	1	DRQ5
1	1	0	DRQ6
1	1	1	DRQ7



10. UIDE CONTROLLER

10.1. INTRODUCTION

An Ultra DMA (UIDE) controller is built into the STPC device. This ATA/IDE hard drive interface protocol doubles the current burst data transfer rate and improves data integrity by using the CRC (Cyclic Redundancy Check) data transfer error detection method. The Ultra DMA (UDMA) protocol is used to transfer data between an Ultra DMA capable IDE controller and one or more Ultra DMA capable IDE devices. It utilizes the standard Bus Master IDE functionality and interface to initiate and control the transfer. The Ultra DMA definition also incorporates a Cyclic Redundancy Checking (CRC-16) error checking protocol. CRC-16 only has the ability for detecting errors, not for correcting them. The current implementation can transfer data using PIO modes and normal single-word and multi-word DMA modes. The Ultra DMA protocol is capable of transferring data at a maximum speed of 66 MBytes/second burst.

10.2. ATA4 COMPLIANT UIDE CONTROLLER

The ATA4 compliant UIDE Controller is embedded into the STPC Vega. This device has the following characteristics:

- UIDE Controller 66
- ATA-4 Industry Standard UIDE controller, for interfacing to devices such as Hard disk drives, CD ROM, etc. The controller supports the following:
 - Support for two connectors to allow the use of up to four drives.
 - Support for CD-ROM and tape peripherals.
 - Support for 11.1/16.6 Mbytes/second, I/O Channel Ready PIO data transfers.
 - Support up to 66 Mbytes/second, UDMA data transfers.
 - Ultra DMA supports CRC-16 error checking protocol (no correction supported)
 - Support for PIO mode 3 & 4 and DMA mode 1 & 2.
 - Backward compatibility with IDE (ATA-1).
 - PCI bus interface.

10.3. UDMA CONTROL REGISTER

This register enables each individual channel and drive for Ultra DMA operation. For non-Ultra DMA operation, this register should be left programmed to its default value.

UDMA_Control				PCI Config F#1			offset 0x49 CPC
7	6	5	4	3	2	1	0
Rsv				SD1 En	SD0 En	PD1 En	PD0 En
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-4	Rsv	Reserved
Bit 3	SD1 En	Secondary Drive 1 UDMA Enable: 1: Enable Ultra DMA mode for secondary channel drive 1 0: Disable (default)
Bit 2	SD0 En	Secondary Drive 0 UDMA Enable: 1: Enable Ultra DMA mode for secondary channel drive 0 0: Disable (default)
Bit 1	PD1 En	Primary Drive 1 UDMA Enable: 1: Enable Ultra DMA mode for primary channel drive 1 0: Disable (default)
Bit 0	PD0 En	Primary Drive 0 UDMA Enable: 1: Enable Ultra DMA mode for primary channel drive 0 0: Disable (default)



10.4. OPERATION

The UDMA protocol is used to transfer data between an Ultra DMA capable IDE controller and one or more Ultra DMA capable IDE devices. It utilizes the standard Bus Master IDE functionality and interface to initiate and control the transfer. The Ultra DMA definition also incorporates a Cyclic Redundancy Checking (CRC-16) error checking protocol. CRC-16 only has the ability for detecting errors, not for correcting them.

10.4.1. Signal Descriptions

The UDMA protocol requires no extra signal pins on the IDE connector, although it does redefine a number of the standard IDE control signals when in Ultra DMA mode. These redefinitions are shown in [Table 10-1](#). Read cycles are defined as transferring data from the IDE device to the IDE Controller. Write cycles are defined as transferring data from the IDE Controller to the IDE device.

Table 10-1. Ultra DMA Control Signal Re-definitions

Standard IDE	UDMA Read Cycle	UDMA Write Cycle	PIO Primary	PIO Secondary
DIOW#	STOP	STOP	PDIOW#	SDIOW#
DIOR#	DMARDY#	STROBE	PDIOR#	SDIOR#
IORDY	STROBE	DMARDY#	PIORDY	SIORDY

The DIOW# signal is redefined as STOP for both read and write transfers. This is always driven by the IDE Controller and is used to request that a transfer be stopped or as an acknowledgment to stop a request from the IDE device.

The DIOR# signal is redefined as DMARDY# for transferring data from the IDE device to the IDE Controller (read). It is used by the IDE Controller to signal when it is ready to transfer data and to add wait states to the current transaction. The DIOR# signal is redefined as STROBE for transferring data from the IDE Controller to the IDE device (write). It is the data strobe signal driven by the IDE Controller on which data is transferred during each rising and falling edge transition.

The IORDY signal is redefined as STROBE for transferring data from the IDE device to the IDE Controller (read). It is the data strobe signal driven by the IDE device on which data is transferred during each rising and falling edge transition. The IORDY signal is redefined as DMARDY# for transferring data from the IDE Controller to the IDE device (write). It is used by the IDE device to signal when it is ready to transfer data and to add wait states to the current transaction. All other signals on the IDE connector retain their functional definitions during Ultra DMA operation.

Initial setup programming consists of enabling and performing the proper configuration of the IDE Controller and the IDE device for Ultra DMA operation. For the IDE Controller, this consists of enabling the Synchronous DMA mode and setting up the appropriate Synchronous DMA timings. When ready to transfer data to or from an IDE device, the Bus Master IDE programming model is followed. Once programmed, the drive and the IDE Controller control the transfer of data via the Ultra DMA protocol. The actual data transfer consists of three phases, a start-up phase, a data transfer phase, and a burst termination phase. The IDE device begins the start-up phase by asserting DMARQ signal. When ready to begin the transfer, the IDE Controller asserts DMACK# signal. When DMACK# signal is asserted, the host controller drives CS0# and CS1# inactive, DA0-DA2 low, and the IDE device drives IOCS16# inactive. For write cycles, the IDE Controller negates STOP, waits for the IDE device to assert DMARDY#, and then drives the first data word and STROBE signal. For read cycles, the IDE Controller tri-states the DD lines, negates STOP, and asserts DMARDY#. The IDE device then sends the first data word and STROBE.

The data transfer phase continues the burst transfers with the data transmitter (the IDE Controller writes, IDE device reads) providing data and toggling STROBE. Data is transferred (latched by receiver) on each rising and falling edge of STROBE. The transmitter can pause the burst by holding STROBE high or low, resuming the burst by again toggling STROBE. The receiver can pause the burst by negating DMARDY# and resumes the transfers by asserting DMARDY#. The IDE Controller pauses a burst transaction to prevent an internal line buffer over or under flow condition, resuming once the condition has cleared. It may also pause a transaction if the current PRD byte count has expired, resuming once it has fetched the next PRD.

The current burst can be terminated by either the transmitter or receiver. A burst termination consists of a Stop Request, Stop Acknowledge and transfer of CRC data. A Burst must first be paused as described above before it can be terminated. The IDE Controller can then stop the burst by asserting STOP, with the IDE device acknowledging by negating DMARQ. The IDE device can then stop the burst by negating DMARQ and the IDE Controller acknowledges by asserting STOP. The transmitter then drives the STROBE signal to a high level. The IDE Controller then drives the CRC value onto the DD lines and negate DMACK#. The IDE device latches the CRC value on rising edge of DMACK#. the IDE Controller terminates a burst transfer if it needs to service the opposite IDE channel, if a Programmed I/O (PIO) cycle is executed to the IDE channel currently running the burst, or upon transferring the last data from the final PRD.

Cyclic Redundancy Checking (CRC-16) is used for error checking on Ultra DMA transfers. The CRC value is calculated for all data by both the IDE Controller and the IDE device over the duration of the Ultra DMA burst transfer segment. This segment is defined as all data transferred with a valid STROBE edge from DDACK# assertion to DDACK# negation. At the end of the transfer burst segment, The IDE Controller drives the CRC value onto the DD[15:0] signals. It is then latched by the IDE device on negation of DDACK#. The IDE device compares the IDE Controller CRC value to its own and reports an error if their is a mismatch.

The timings for Ultra DMA are programmed into the Ultra DMA Timing Register. The programmable timings include Cycle Time (CT) and Ready to Pause (RP) time. The Cycle Time represents the minimum pulse width of active data strobe (STROBE) signal. The Ready to Pause time represents the number of PCI clocks the IDE Controller waits from negation of DMARDY# to the assertion of STOP when it desires to stop a burst read transaction.

10.5. IDE CONTROLLER OPERATION

The IDE (Integrated Drive Electronics) controller provides two IDE channels, primary and secondary, for interfacing with up to four IDE drives. It supports PIO modes 0 to 4 plus DMA modes 0 to 2. The timings are individually programmable for all four IDE devices. Each channel has a four double-word FIFO for data transfers which allows four levels of write posting or read prefetch. Accesses to the 8-bit non-data IDE registers bypass the FIFOs.

For each of the four drives there are three bits in the configuration registers which can selectively enable write posting, read prefetch and ATAPI read prefetch. If read prefetch is enabled, the IDE controller will prefetch data from the drive after the first read has been made. The prefetching will stop after 256 data reads (512 Bytes), which is the normal sector size. If the current command to the drive is ATAPI packet (A0h), or service (A2h), then the read prefetch will be disabled unless ATAPI read prefetch is set.

The two channels of the IDE controller can be individually programmed to operate in either legacy or native mode. In legacy mode, the IDE interrupts are hardwired to INT 14 & 15. In native mode, they both connect to PCI INTA. If legacy mode is selected, INT 14 & 15 will not be available on the ISA bus even if IDE interrupts are disabled. In legacy mode, the primary and secondary channels are hardwired to IO addresses 1F0h-1F7h and 170h-177h respectively and also 376h for both channels. In native mode the IO addresses are programmed by configuration registers. For information on PIO mode, please refer to the ATAPI Standard.

The IDE controller provides DMA bus master transfer between IDE devices and system memory, with scatter/gather capability. By performing the IDE data transfer as a bus master, the Bus Master Device offloads the CPU (no programmed IO for data transfer) and improves system performance in multitasking environments.

Before issuing the DMA command, the system software must first create a Physical Region Descriptor (PRD) table in system memory. This table contains a list of pointers and byte counts for each entry. A register in the IDE controller is set to point to this table. The IDE DMA controller will read from system memory during DMA initialization. Each entry in the PRD table is eight bytes long and will have the format below:

10.6. PRD Table Entry

PRD1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EOT	Rsv														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOW															Rsv
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31	OET	This bit set to '1' if it is the last entry in the table
Bits 30-16	Rsv	Reserved
Bit 15-1	NOWS	Number of 16-bit data packets
Bit 0	Rsv	Reserved; This bit must be set to 0

PRD0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MRPAS															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRPAS															Rsv
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bit 31-1	MRPAS	Memory Physical Address of the first descriptor
Bit0	Rsv	Reserved; This bit must be set to 0

The table must be aligned on a 4 byte boundary and should not cross a 64k boundary.

A memory region also should not cross a 64k boundary neither. An example of a PRD table is shown in [Figure 10-1..](#)

The primary and secondary channels each have a PRD address pointer register.

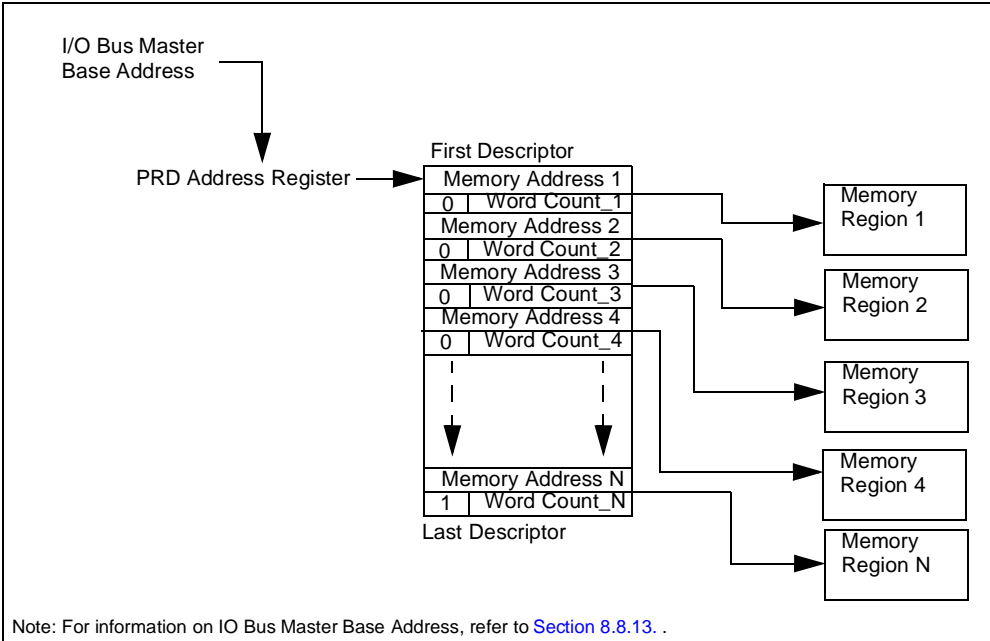


Figure 10-1. PRD Table Entry Example

To save pins, in the ISA mode, the IDE controller shares pins with the ISA interface. On the IDE data bus, CS1 & CS3 signals are shared with the ISA address bus and keyboard controller/RTC pins. These signals are isolated by external transceiver devices. The ISAOE signal selects whether the pins are in IDE or ISA mode. The South Bridge arbitrates between the IDE controller and the ISA bus bridge to select which has control of the shared pins. Note that for the Local Bus mode, the IDE pins are not multiplexed.

10.7. IDE Bus Master Registers

This document defines a register level programming interface for the internal busmaster ATA-compatible (IDE) disk controller that directly moves data between IDE devices and main memory.

The system using this programming interface will benefit from bundled software shipped with major Operating Systems, limiting the amount of software development required to provide a complete product.

The master mode programming interface is an extension of the standard IDE programming model. This means that devices can always be dealt with using the standard IDE programming model, with the master mode functionality used when the appropriate driver and devices are present. Master operation is designed to work with any IDE device that supports DMA transfers on the IDE bus. Devices that only work in PIO mode can be used through the standard IDE programming model.

The programming interface defines a simple scatter/gather mechanism, allowing large transfer blocks to be scattered to or gathered from memory. This cuts down the number of interrupts to and interactions with the CPU. The interface defined here supports two IDE channels (primary and secondary).

10.7.1. Physical Region Descriptor Table

Before the controller starts a master transfer, it is given a pointer to a Physical Region Descriptor Table. This table contains some of a number of the Physical Region Descriptors (PRD); these define the memory

areas that are involved in the data transfer. The descriptor table must be aligned on a 4 Byte boundary and the table cannot cross a 64 KByte boundary in memory.

10.7.2. Physical Region Descriptor

The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The data transfer will not proceed until all regions described by the PRDs in the table have been transferred.

Each Physical Region Descriptor entry is eight Bytes long.

- The first four bytes specify the byte address of a physical memory region.
- The next two bytes specify the count of the region in bytes (64K byte limit per region).

A value of zero in these two bytes indicates 64 KByte. Bit 7 of the last byte indicates the end of the table; the Bus Master operation terminates when the last descriptor has been retired.

Note: The memory region specified by the descriptor is further restricted such that the region cannot straddle a 64K boundary. This means that the byte count can be limited to 64K, and the incrementer for the current address register need only extend from bit [1] to bit [15]. Also, the total sum of the descriptor byte counts must be equal to, or greater than, the size of the disk transfer request. If greater than, then the driver must terminate the Bus Master transaction (by resetting bit zero of the command register to zero) when the drive issues an interrupt to signal transfer completion.

10.8. Bus Master IDE Register Description

The bus master IDE function uses 16 bytes of IO space. All bus master IDE IO space registers can be accessed as byte, word or Dword quantities. The description of the 16 bytes of IO registers is given in [Table 10-2](#).

Table 10-2. Bus Master IDE Register Description

Offset	Register	R/W Status
00h	Bus Master IDE Command register Primary	R/W
01h	Device Specific	
02h	Bus Master IDE Status register Primary	RWC
03h	Device Specific	
04h-07h	Bus Master IDE PRD Table Address Primary	R/W
08h	Bus Master IDE Command register Secondary	R/W
09h	Device Specific	
0Ah	Bus Master IDE Status register Secondary	RWC
0Bh	Device Specific	
0Ch-0Fh	Bus Master IDE PRD Table Address Secondary	R/W



10.9. Bus Master IDE Command Register

10.9.1. IDE Command Register

This 8-bit Register is addressed at offset Base + 00h for the Primary IDE Channel and Base + 08h for the Secondary IDE Channel.

This register enables/disables Bus Master capability for the IDE function and provides direction control for the IDE DMA transfers. This register also provides the bits that software uses to indicate DMA capability of the IDE device.

IDE_COM

7	6	5	4	3	2	1	0
Rsv				RWCOM	Rsv		SSBM
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7-4	Rsv	Reserved
Bit 3	RWCOM	<p>Read Write Control. This bit sets the direction of the bus master transfer: when set to zero, PCI bus master reads are performed. When set to one, PCI bus master writes are performed:</p> <p>0 = PCI bus master read 1 = PCI bus master write</p> <p>While a synchronous DMA transfer is in progress, this bit will be READ ONLY. The bit will return to read/write once the synchronous DMA transfer has been completed or halted.</p> <p>This bit must NOT be changed when the bus master function is active</p>
Bit 2-1	Rsv	Reserved
Bits 0	SSBM	<p>Stop/Start Bus Master. Writing a '1' to this bit enables bus master operation of the controller. Bus master operation begins when this bit is detected changing from a zero to a one. The controller will transfer data between the IDE device and memory only when this bit is set. Master operation can be halted by writing a '0' to this bit. All state information is lost when a '0' is written;. Master mode operation cannot be stopped and then resumed. If this bit is reset while bus master operation is still active (i.e., Bit 0= 1 in the Bus Master IDE Status Register for that IDE channel) and the drive has not yet finished its data transfer (bit 2=0 in the channel's Bus Master IDE Status Register), the Bus Master command is said to be aborted and data transferred from the drive may be discarded before being written to the system memory. This bit is intended to be reset after the data transfer is completed, as indicated by either the Bus Master IDE Active bit or the Interrupt bit of the IDE Status register for that IDE channel being set, or both.</p>

10.9.2. IDE Status Register

This 8-bit Register is addressed at offset Base + 02h for the Primary IDE Channel and Base + 0Ah for the Secondary IDE Channel.

This register provides status information about the IDE device and state of the IDE DMA transfer. [Table 10-3.](#) describes IDE Interrupt Status and Bus Master IDE Active bit states after a DMA transfer has been started.

The IDE Status Register is illustrated in the following table.

IDE_COM

7	6	5	4	3	2	1	0
SO	D1DMA	D0DMA	Rsv		RWI	RWE	RWMI
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	SO	Simplex only. This is hardwired to '0'.
Bit 6	D1DMA	Drive 1 DMA Capable. This read/write bit is set by device dependent code (BIOS or device driver) to indicate that drive 1 for this channel is capable of DMA transfers, and that the controller has been initialised for optimum performance.
Bit 5	D0DMA	Drive 0 DMA Capable. This read/write bit is set by device dependent code (BIOS or device driver) to indicate that drive 0 for this channel is capable of DMA transfers, and that the controller has been initialised for optimum performance.
Bits 4-3	Rsv	Reserved. These bits return '0' when read.
Bit 2	RWI	Read/Write Interrupt. This bit is set by the rising edge of the IDE interrupt line. This bit is cleared when a '1' is written to it by software. Software can use this bit to determine if an IDE device has asserted its interrupt line. When this bit is read as a one, all data transferred from the drive is visible in system memory. For further details see Table 10-3.
Bit 1	RWE	Read/Write Error. This bit is set when the controller encounters an error in transferring data to/from memory. The exact error condition is bus specific and can be determined in a bus specific manner. This bit is cleared when a '1' is written to it by software.
Bit 0	RWMI	Read/Write Bus Master IDE Active. This bit is set to 1 when bit 0 in the Command register is set to 1. This bit is cleared (set to 0) when the last transfer for a region is performed, where EOT for that region is set in the region descriptor. It is also cleared when the Start bit is cleared in the Command register. When this bit is read as a zero, all data transferred from the drive during the previous bus master command is visible in system memory, unless the bus master command was aborted. For further details see Table 10-3.



Table 10-3. Interrupt/Activity Status Combinations

Bit 2	Bit 0	Description
0	1	DMA transfer is in progress. No interrupt has been generated by the IDE device.
1	0	The IDE device generated an interrupt and the Physical Region Descriptors exhausted. This is normal completion where the size of the physical memory regions is equal to the IDE device transfer size.
1	1	The IDE device generated an interrupt. The controller has not reached the end of the physical memory regions. This is a valid completion case when the size of the physical memory regions is larger than the IDE device transfer size.
0	0	Error condition. If the IDE DMA Error bit is 1, there is a problem transferring data to/from

10.9.3. Descriptor Table Pointer Register

This 32-bit Register is addressed at offset Base I/O+ 04h for the Primary IDE Channel and Base I/O+ 0Ch for the Secondary IDE Channel.

This register provides the base memory address of the Descriptor Table. The Descriptor Table must be DWord aligned and must not cross a 4-Kbyte boundary in memory.

DT_Point

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BADT															
Default value after reset = 00h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BADT														Rsv	
Default value after reset = 00h															

Bit Number	Mnemonic	Description
Bits 31-2	BADT	Base address of Descriptor table. This field corresponds to A[31-2].
Bits 1-0	Rsv	Reserved

The Descriptor Table must be Dword aligned. The Descriptor Table must not cross a 64K boundary in memory.



10.10. Operation

10.10.1. Standard Programming Sequence

To initiate a bus master transfer between memory and an Hard Disk device, the following steps are required:

- 1) Software prepares a PRD Table in system memory. Each PRD is 8 bytes long and consists of an address pointer to the starting address and the transfer count of the memory buffer to be transferred. In any given PRD Table, two consecutive PRDs are offset by 8-bytes and are aligned on a 4-byte boundary.
- 2) Software provides the starting address of the PRD Table by loading the PRD Table Pointer Register. The direction of the data transfer is specified by setting the Read/Write Control bit. Clear the Interrupt bit and Error bit in the Status register.
- 3) Software issues the appropriate DMA transfer command to the disk device.
- 4) Engage the bus master function by writing a '1' to the Start bit in the Bus Master IDE Command Register for the appropriate channel.
- 5) The controller transfers data to/from memory responding to DMA requests from the IDE device.
- 6) At the end of the transfer the IDE device signals an interrupt.
- 7) In response to the interrupt, software resets the Start/Stop bit in the command register. It then reads the controller status and then the drive status to determine if the transfer completed successfully.

10.11. Data Synchronization

When reading data from an IDE device, that data may be buffered by the IDE controller before using a master operation to move the data to memory. The IDE device driver in conjunction with the IDE controller is responsible for guaranteeing that any buffered data is moved into memory before the data is used.

The IDE device driver is required to do a read of the controller Status register after receiving the IDE interrupt. If the Status register returns with the Interrupt bit set, then the driver knows that the IDE device generated the interrupt (important for shared interrupts) and that any buffered data has been flushed to memory. If the Interrupt bit is not set, then the IDE device did not generate the interrupt and the state of the data buffers is unknown.

When the IDE controller detects a rising edge on the IDE device interrupt line (INTRQ), it is required to:

- Flush all buffered data
- Set the Interrupt bit in the controller Status register
- Guarantee that a read to the controller Status register does not complete until all buffered data has been written to memory.

Another way to view this requirement is that the first read to the controller Status register in response to the IDE device interrupt must return with the Interrupt bit set and with the guarantee that all buffered data has been written to memory.

10.11.1. Status Bit Interpretation

The table below gives a description of how to interpret the Interrupt and Active bits in the Controller status register after a DMA transfer has been started.

Interrupt bit	Active bit	Description:
0	1	DMA transfer is in progress. No interrupt has been generated by the IDE device.
1	0	The IDE device generated an interrupt. The controller exhausted the Physical Region Descriptors. This is the normal completion case where the size of the physical memory regions was equal to the IDE device transfer size.
1	1	The IDE device generated an interrupt. The controller has not reached the end of the physical memory regions. This is a valid completion case where the size of the physical memory regions was larger than the IDE device transfer size.
0	0	This bit combination signals an error condition. If the Error bit in the status register is set, then the controller has some problem transferring data to/from memory. Specifics of the error have to be determined using bus-specific information. If the Error bit is not set, then the PRD's specified a smaller size than the IDE transfer size.

10.12. Error Conditions

IDE devices are sector based mass storage devices. The drivers handle errors on a sector by sector basis; either a sector is transferred successfully or it is not. If the IDE DMA slave device never completes the transfer due to a hardware or software error, the Bus Master IDE command will eventually be stopped (by setting Command Start bit to zero) when the driver times out the disk transaction. Information in the IDE device registers will help isolate the cause of the problem.

If the controller encounters an error while doing the bus master transfers, it will stop the transfer (i.e. reset the Active bit in the Command register) and set the ERROR bit in the Status register. The controller does not generate an interrupt when this happens. The device driver can use device specific information (e.g. PCI Configuration Space Status register) to determine what caused the error.

Whenever a requested transfer does not complete properly, information in the IDE device registers (Sector Count) can be used to determine how much of the transfer was completed and to construct a new PRD table to complete the requested operation. In most cases the existing PRD table can be used to complete the operation.

10.13. PCI Specifics

Bus master IDE controllers built to attach to a PCI bus must have the following characteristics:

- 1) The Class Code in PCI configuration space indicates IDE device and bit 7 of the Programming Interface register (offset 0x09) in PCI configuration space must be set to 1 to indicate that the device supports the Master IDE capability.
- 2) The control registers for the controller are allocated via the devices Base Address register at offset 0x20 in PCI configuration space.
- 3) In the controller Status register the Error bit will be set and the Active bit reset if any of the following conditions occur on the PCI bus while the controller is doing a master operation on the bus. The exact cause can be determined by examining the Configuration Space Status register.



Error Condition	Configuration Space Status bits
Target Abort	Any time bit 12 of the Config Space Status register is set.
Master Abort	Any time bit 13 of the Config Space Status register is set.
Data Parity	Any time bit 8 of the Config Space Status register is set.
Error Detected	



11. LOCAL BUS INTERFACE

11.1. INTRODUCTION

The Local Bus interface of the STPC provides a low latency bus to external peripheral. The Local Bus may operate the 25-bit address and 16-bit data bus.

The Local Bus interface supports up to two memory banks and four I/O devices. It can support up to 16 MBytes of memory in each memory bank and from 1 Byte to 1 KByte of I/O space for each of the I/O devices. All the chip select timings are individually programmable. This interface can be accessed only by the CPU.

The first bank of the memory is intended to be used as the boot device.

The starting address for each I/O chip select is programmable at the 1 Byte boundary. The access starting range for each of the chip selects is also programmable. The size varies from 1 Byte to 1 KByte.

Note: The base address of any local bus slot (I/O or memory) has to be an integral multiple of the size of the slot. For example, a 32 MByte flash must have a 32 MByte aligned base address.

11.1.1. FEATURES

- Support of 8/16/32-bit cycles for both 8/16-bit I/O or memory devices
- Two banks of 16 MByte (max.) each, one as a boot device
- Memory Banks size and programmable base address
- Programmable timing with host clock granularity for Bank and I/O accesses
- Up to four I/O devices supported with programmable start address & size
- I/O device timing (setup & recovery time) programmable
- Interrupt support

11.2. MEMORY BANK SWITCHING

The Local Bus Interface caters for two memory banks of up to 16 MBytes (max), designated Bank 0 (upper) and Bank 1 (lower).

FCS0#: Chip Select for Bank 0

FCS1#: Chip Select for Bank 1

The base addresses for the two Banks are specified in registers MEMAREG0 ([Section 11.5.6.](#)) and MEMAREG1 ([Section 11.5.7.](#)). The size of memory allocated to each of the two Banks is specified in the MEMMASK register ([Section 11.5.8.](#)). Note that the minimum size that can be allocated to a Bank is 1 MByte.

11.3. FLASH DEVICE IMPLEMENTATION

The Local Bus Controller is capable of managing two 16MByte flash devices in 8 or 16bit wide configuration allowing great flexibility of implimentation.

The choice of configuration depends on the boot method required;

11.3.1. STANDARD BIOS BOOT OR BOOT LOADER IN REAL MODE

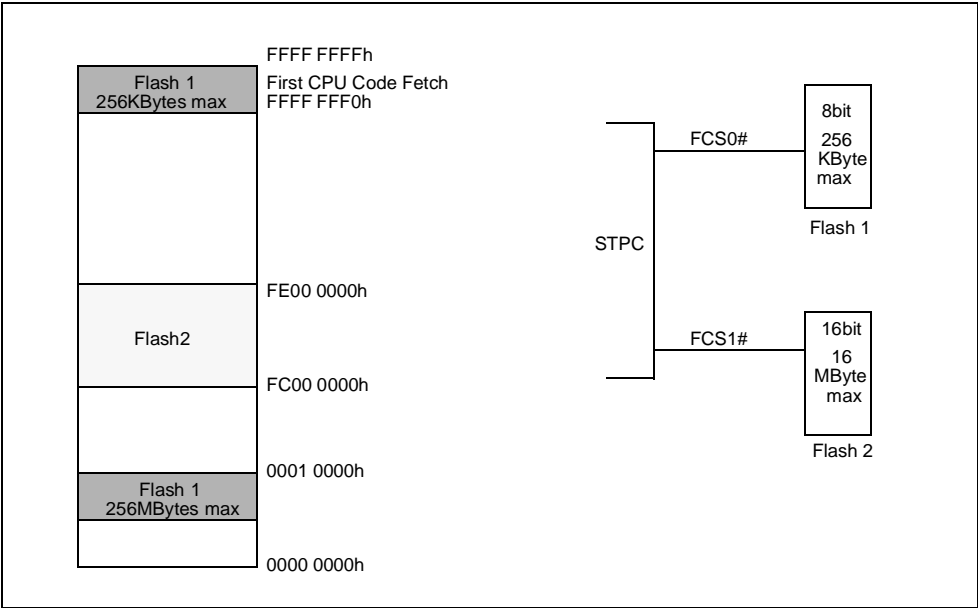
In order to execute boot code below the 1st MByte of memory, the flash device must imperatively be connected to FCS0#, 8 bit wide and have a maximum capacity of 256 KBytes. In this case, bank 1 is used in the same as an ISA bus implementation.

Bank 1 may be used to store data or an OS which is loaded by the code stored in bank 0.

The limitation of this implimentation is that only 16MBytes are available for Data and OS storage and booting in 8bit configuration is available.

This is illustrated in [Figure 11-1.](#)

Figure 11-1. Standards BIOS Boot Illustration



11.3.2. BOOTLOADER THAT IS EXECUTED ABOVE THE FIRST MBYTE

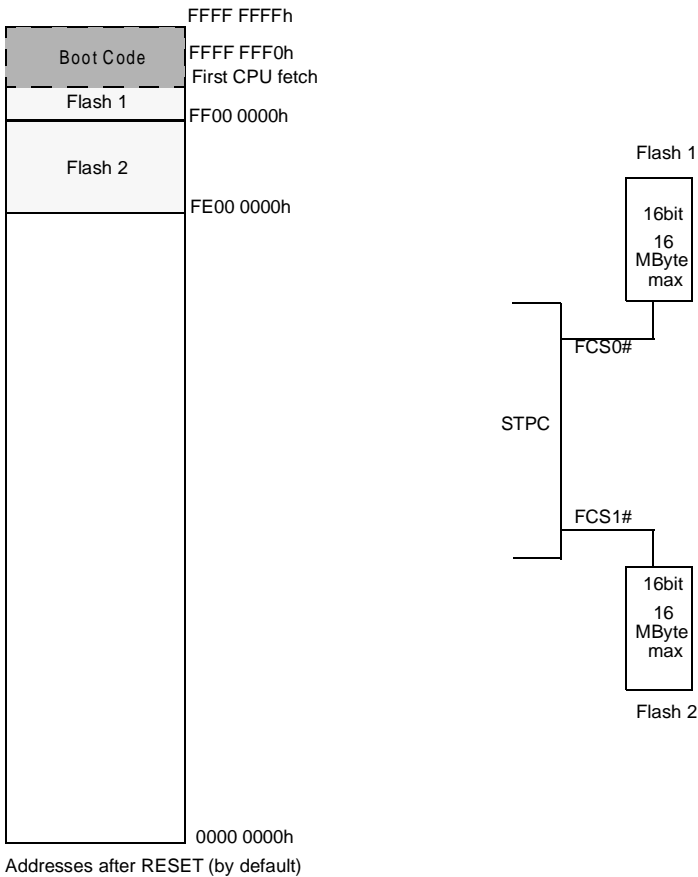
In order to use high capacity 16bit flash devices on Bank 0, the boot code must be executed above the 1MByte. The first CPU fetch is carried out at FFFF FFF0h (4GByte - 16Byte).

The CPU is only able to see flash devices that are located above 1MBytes of memory. Therefore, the boot code needs to initialise the memory and copy itself below the first MByte before it is able to use a stack, make function calls or to modify the Code Segment Register CS (which is equivalent of reading the next CPU instructions under the first MByte).

To simplify the developer's task, a primary bootloader has been developed by STMicroelectronics. This bootloader initialises the STPC, detects and configures the memory before being copied below the first MByte, and hands over to next boot programme (Secondary loader).

This is illustrated in [Figure 11-2](#).

Figure 11-2. Two flash device implimentation



Note; In this case the Bank size can't be modified and the base address can only be changed at a Modulo of 32 MBytes

11.4. CONFIGURATION REGISTERS

The Local Bus configuration registers can be categorised into five groups:

- 1) Base Index Register
- 2) Address Decode Registers
- 3) Timing Registers
- 4) Control Register
- 5) Device Width Register

Table 11-1. 16-bit Address Decode Registers for I/O and MEM

Offset	Register Name	Register Description	Default Values
Variable, see Section 11.5 .	BASE_INDEX_REG[15:0]	<i>Base_Index_Reg</i> for Configuration registers	Base
0x0000h	IOAREG0[15:0] Base IO 0	Base Address of I/O Slots 0	0xFFFFh
0x0002h	IOAREG1[15:0] Base IO 1	Base Address of I/O Slots 1	0xFFFFh
0x0004h	IOAREG2[15:0] Base IO 2	Base Address of I/O Slots 2	0xFFFFh
0x0006h	IOAREG3[15:0] Base IO 3	Base Address of I/O Slots 3	0xFFFFh
0x0008h	IOAREG4[15:0] Base IO 4	Base Address of I/O Slots 4	0xFFFFh
0x000Ah	IOAREG5[15:0] Base IO 5	Base Address of I/O Slots 5	0xFFFFh
0x000Ch	IOAREG6[15:0] Base IO 6	Base Address of I/O Slots 6	0xFFFFh
0x000Eh	IOAREG7[15:0] Base IO 7	Base Address of I/O Slots 7	0xFFFFh
0x0010h	IOMREG0[15:0] Mask IO0	Mask Size of I/O Slots 0	0xFFFFh
0x0012h	IOMREG1[15:0] Mask IO1	Mask Size of I/O Slots 1	0xFFFFh
0x0014h	IOMREG2[15:0] Mask IO2	Mask Size of I/O Slots 2	0xFFFFh
0x0016h	IOMREG3[15:0] Mask IO3	Mask Size of I/O Slots 3	0xFFFFh
0x0018h	IOMREG4[15:0] Mask IO4	Mask Size of I/O Slots 4	0xFFFFh
0x001Ah	IOMREG5[15:0] Mask IO5	Mask Size of I/O Slots 5	0xFFFFh
0x001Ch	IOMREG6[15:0] Mask IO6	Mask Size of I/O Slots 6	0xFFFFh
0x001Eh	IOMREG7[15:0] Mask IO7	Mask Size of I/O Slots 7	0xFFFFh

Table 11-1. 16-bit Address Decode Registers for I/O and MEM

Offset	Register Name	Register Description	Default Values
0x0020h	TIMEMEM0[15:0]	Timing Template for accessing bank 0	0xA87Dh
0x0022h	TIMEMEM1[15:0]	Timing Template for accessing bank 1	0xA87Dh
0x0024h	TIMEIO0[15:0]	Timing Template for accessing I/O 0	0x0000h
0x0026h	TIMEIO1[15:0]	Timing Template for accessing I/O 1	0x0000h
0x0028h	TIMEIO2[15:0]	Timing Template for accessing I/O 2	0x0000h
0x002Ah	TIMEIO3[15:0]	Timing Template for accessing I/O 3	0x0000h
0x002Ch	TIMEIO4[15:0]	Timing Template for accessing I/O 4	0x0000h
0x002Eh	TIMEIO5[15:0]	Timing Template for accessing I/O 5	0x0000h
0x0030h	TIMEIO6[15:0]	Timing Template for accessing I/O 6	0x0000h
0x0032h	TIMEIO7[15:0]	Timing Template for accessing I/O 7	0x0000h
0x0034h	CONTROL[15:0]	16-bit Control Register	0x0001h
0x0036h	DEVICE_WIDTH[15:0]	I/O or MEM width as 8-bit or 16-bit	0x0300h
0x0038h	MEMAREG0[15:0]	Base address [31:20]of Memory bank0	0x0FE0h
0x003Ah	MEMAREG1[15:0]	Base address [31:20]of Memory bank1	0x0FC0h
0x003Ch	MEMMASK[15:0]	Address range of Memory bank0 & 1	0x003Fh

11.5. LOCAL BUS BASE INDEX REGISTER

The Atlas local bus base address is variable. Access can be calculated using the Pseudo code given below. The following assumes that the Local Bus (LB) is device number 6, and that the LB I/O index is 28h with the data at 2Ch.

11.5.1. INITIALISATION

Select the Local Bus device:

```
IOWRITE8 (22h, 10h)
IOWRITE8 (23h, 06h)
IOWRITE8 (22h, 11h)
IOWRITE8 (23h, 00h)
```

Set the Local Bus base address and enable access:

```
IOWRITE8 (22h, 12h)
IOWRITE8 (23h, 28h)
IOWRITE8(23h,(HOST_BASE &FFh) | 03h);
IOWRITE8 (22h, 13h)
IOWRITE8 (23h, 00h)
IOWRITE8(23h,HOST_BASE >>8);
```

11.5.2. WRITE ACCESS

```
IOWRITE16 (HOST_BASE, register_index)
IOWRITE16 (HOST_BASE + 4, data)
```

11.5.3. READ ACCESS

```
IOWRITE16 (HOST_BASE, register_index)
Data = IOREAD16 (HOST_BASE + 4)
```

Note that further information is available in [Section 6.8](#).

11.5.4. I/O SLOT BASE ADDRESS REGISTERS IOAREG0 TO IOAREG7

IOAREG#																Access = see Section 11.5 .				Regoffset: see Table 11-1			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
SA																							
Default value after reset = FFFFh																							

Bit Number	Mnemonic	Description
Bits 15-0	SA	Starting Address aligned to a Byte.



11.5.5. I/O SLOT MASK REGISTERS

The eight address mask registers IOMREG0 to IOMREG7 define the size of each I/O slot. The 10-bit address mask will mask the part of the address that is not to be used during the address decoding process. The 10-bit mask will filter bit 9:0 of the starting address specified in the corresponding IOAREG slot base register.

IOMREG#																Access = see Section 11.5 .	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Rsv						AMIO											
Default value after reset = FFFFh																	

Bit Number	Mnemonic	Description
Bits 15-10	Rsv	Reserved
Bits 9-0	AMIO	Bits used for address masking.

Register	Regoffset
IOMREG0	10h
IOMREG1	12h
IOMREG2	14h
IOMREG3	16h
IOMREG4	18h
IOMREG5	1Ah
IOMREG6	1Ch
IOMREG7	1Eh



11.5.6. MEMORY BASE ADDRESS REGISTER 0

The Base Address for Bank0 is specified by base address register MEMAREG0. The minimum size that a bank can have is 1Mb.

MEMAREG0				Access = see Section 11.5 .											Regoffset = 38h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Rsv																
Default value after reset = 0FE0h																

Bit Number	Mnemonic	Description
Bits 15-12	Rsv	Reserved
Bits 11-0		Bits 31:20 of memory base for Bank0

Bank0 has a Base Address register mapped at 0xFE000000h after reset, and having a size of 32Mb (max).



11.5.7. MEMORY BASE ADDRESS REGISTER 1

The Base Address for Bank1 is specified by base address register MEMAREG1. The minimum size that a bank can have is 1Mb.

MEMAREG1				Access = see Section 11.5.								Regoffset = 3Ah			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv															
Default value after reset = 0FC0h															

Bit Number	Mnemonic	Description
Bits 15-12	Rsv	Reserved
Bits 11-0		Bits 31:20 of memory base for Bank1

Note: The Base for Bank1 should be anything above the physical DRAM address and up to anything below (0xFE000000 - size of Bank1).

For example, for a Base Address of 0xF0000000h for bank1, MEMAREG1 = 0F00h.



11.5.8. MEMORY MASK REGISTER

The MEMMASK register defines the memory sizes for Bank0 and Bank1.

MEMMASK															Access = see Section 11.5.					Regoffset = 3Ch				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
Rsv								Rsv																
Default value after reset = 003Fh																								

Bit Number	Mnemonic	Description
Bits 15-14	Rsv	Reserved
Bits 13-9		Mask of Memory Bank1 in Mb
Bit 8		Enable/Disable Memory Bank1 0 = disable, 1 = enable
Bits 7-6	Rsv	Reserved
Bits 5-1		Mask of Memory Bank0 in Mb
Bit 0		Enable/Disable Memory Bank0 0 = disable, 1 = enable

Note: It is recommended that the Mask should be specified in powers of two -1, as in the following Table, so that the memory size could be 1Mb, 2 Mb, 4 Mb, 8 Mb, 16 Mb or 32 Mb.

Coding Bits 13:9 (Bank1) & 5:1 (Bank0)					Mask Size
0	0	0	0	0	1 Mb
0	0	0	0	1	2 Mb
0	0	0	1	1	4 Mb
0	0	1	1	1	8 Mb
0	1	1	1	1	16 Mb
1	1	1	1	1	32 Mb
Any other value not mentioned in the above is forbidden					

The default value (001Fh) represents 32 Mb max. and bank0 enabled, bank1 disabled.



11.6. LOCAL BUS TIMING REGISTERS

11.6.1. TIMING MEMORY TEMPLATE REGISTER 0

TIMEMEM0 defines the timing template for accessing Flash memory Bank0. The timing is programmed with reference to the host clock period as a time unit.

TIMEMEM0

Access = see [Section 11.5.](#)

Regoffset = 20h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSH	UARS	CHT			CAT							CST			
Default value after reset = A87Dh															

Bit Number	Mnemonic	Description
Bit 15	DSH	Discard the Setup and Hold parameters for read operations. 0 = Do not discard, 1 = discard
Bit 14	UARS	Use asynchronous ready signal for command termination. The asynchronous ready enable bit sets the Local Bus logic to wait for an external ready signal before closing the current cycle. 0 = Do not use, 1 = Use
Bits 13-11	CHT	Command hold time , and is determined as follows: Command hold time = (4+Vh) x T where Vh = Register value for the Hold time T = HCLK period
Bits 10-3	CAT	Command Active time , and is determined as follows: Command active time = (2+Va) x T where Va = Register value for the Active time T = HCLK period.
Bits 2-0	CST	Command Setup time , and is determined as follows: Command setup time = (4+Vs) x T where Vs = Register value for the Setup time T = HCLK period.

11.6.2. TIMING MEMORY TEMPLATE REGISTER 1

TIMEMEM1 defines the timing template for accessing Flash memory Bank1. The timing is programmed with reference to the host clock period as a time unit.

TIMEMEM1

Access = see [Section 11.5](#).

Regoffset = 22h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSH	UARS	CHT			CAT								CST		
Default value after reset = A87Dh															

Bit Number	Mnemonic	Description
Bit 15	DSH	Discard the Setup and Hold parameters for read operations. 0 = Do not discard, 1 = discard
Bit 14	UARS	Use asynchronous ready signal for command termination. The asynchronous ready enable bit sets the Local Bus logic to wait for an external ready signal before closing the current cycle. 0 = Do not use, 1 = Use
Bits 13-11	CHT	Command hold time , and is determined as follows: Command hold time = $(4+V_h) \times T$ where V_h = Register value for the Hold time T = HCLK period
Bits 10-3	CAT	Command Active time , and is determined as follows: Command active time = $(2+V_a) \times T$ where V_a = Register value for the Active time T = HCLK period.
Bits 2-0	CST	Command Setup time , and is determined as follows: Command setup time = $(4+V_s) \times T$ where V_s = Register value for the Setup time T = HCLK period.

11.6.3. I/O TIMING TEMPLATE REGISTER 0

TIMEIO0 defines the timing template for accessing device in I/O Slot 0. The timing is programmed with reference to the host clock period as a time unit.

TIMEIO0

Access = see [Section 11.5](#).

Regoffset = 24h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSH	UARS	CHT			CAT								CST		
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bit 15	DSH	Discard the Setup and Hold parameters for read operations. 0 = Do not discard, 1 = discard
Bit 14	UARS	Use Asynchronous Ready Signal for command termination. The asynchronous ready enable bit sets the Local Bus logic to wait for an external ready signal before closing the current cycle. 0 = Do not use, 1 = Use
Bits 13-11	CHT	Command Hold Time , and is determined as follows: Command hold time = (4+Vh) x T where Vh = Register value for the Hold time T = HCLK period
Bits 10-3	CAT	Command Active Time , and is determined as follows: Command active time = (2+Va) x T where Va = Register value for the Active time T = HCLK period.
Bits 2-0	CST	Command Setup Time , and is determined as follows: Command setup time = (4+Vs) x T where Vs = Register value for the Setup time T = HCLK period.

11.6.4. I/O TIMING TEMPLATE REGISTER 1

TIMEIO1 defines the timing template for accessing device in I/O Slot 1. The timing is programmed with reference to the host clock period as a time unit.

TIMEIO1

Access = see [Section 11.5](#).

Regoffset = 26h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSH	UARS	CHT			CAT								CST		
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bit 15	DSH	Discard the Setup and Hold parameters for read operations. 0 = Do not discard, 1 = discard
Bit 14	UARS	Use Asynchronous Ready Signal for command termination. The asynchronous ready enable bit sets the Local Bus logic to wait for an external ready signal before closing the current cycle. 0 = Do not use, 1 = Use
Bits 13-11	CHT	Command Hold Time , and is determined as follows: Command hold time = $(4+V_h) \times T$ where V_h = Register value for the Hold time T = HCLK period
Bits 10-3	CAT	Command Active Time , and is determined as follows: Command active time = $(2+V_a) \times T$ where V_a = Register value for the Active time T = HCLK period.
Bits 2-0	CST	Command Setup Time , and is determined as follows: Command setup time = $(4+V_s) \times T$ where V_s = Register value for the Setup time T = HCLK period.

11.6.5. I/O TIMING TEMPLATE REGISTER 2

TIMEIO2 defines the timing template for accessing device in I/O Slot 2. The timing is programmed with reference to the host clock period as a time unit.

TIMEIO2															Access = see Section 11.5 .			Regoffset = 28h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
DSH	UARS	CHT			CAT								CST							
Default value after reset = 0000h																				

Bit Number	Mnemonic	Description
Bit 15	DSH	Discard the Setup and Hold parameters for read operations. 0 = Do not discard, 1 = discard
Bit 14	UARS	Use Asynchronous Ready Signal for command termination. The asynchronous ready enable bit sets the Local Bus logic to wait for an external ready signal before closing the current cycle. 0 = Do not use, 1 = Use
Bits 13-11	CHT	Command Hold Time , and is determined as follows: Command hold time = (4+Vh) x T where Vh = Register value for the Hold time T = HCLK period
Bits 10-3	CAT	Command Active Time , and is determined as follows: Command active time = (2+Va) x T where Va = Register value for the Active time T = HCLK period.
Bits 2-0	CST	Command Setup Time , and is determined as follows: Command setup time = (4+Vs) x T where Vs = Register value for the Setup time T = HCLK period.



11.6.6. I/O TIMING TEMPLATE REGISTER 3

TIMEIO3 defines the timing template for accessing device in I/O Slot 3. The timing is programmed with reference to the host clock period as a time unit.

TIMEIO3

Access = see [Section 11.5](#).

Regoffset = 2Ah

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSH	UARS	CHT			CAT								CST		
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bit 15	DSH	Discard the Setup and Hold parameters for read operations. 0 = Do not discard, 1 = discard
Bit 14	UARS	Use Asynchronous Ready Signal for command termination. The asynchronous ready enable bit sets the Local Bus logic to wait for an external ready signal before closing the current cycle. 0 = Do not use, 1 = Use
Bits 13-11	CHT	Command Hold Time , and is determined as follows: Command hold time = $(4+V_h) \times T$ where V_h = Register value for the Hold time T = HCLK period
Bits 10-3	CAT	Command Active Time , and is determined as follows: Command active time = $(2+V_a) \times T$ where V_a = Register value for the Active time T = HCLK period.
Bits 2-0	CST	Command Setup Time , and is determined as follows: Command setup time = $(4+V_s) \times T$ where V_s = Register value for the Setup time T = HCLK period.

11.6.7. I/O TIMING TEMPLATE REGISTER 4

TIMEIO4 defines the timing template for accessing device in I/O Slot 4. The timing is programmed with reference to the host clock period as a time unit.

TIMEIO4										Access = see Section 11.5 .					Regoffset = 2Ch				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
DSH	UARS	CHT			CAT							CST							
Default value after reset = 0000h																			

Bit Number	Mnemonic	Description
Bit 15	DSH	Discard the Setup and Hold parameters for read operations. 0 = Do not discard, 1 = discard
Bit 14	UARS	Use Asynchronous Ready Signal for command termination. The asynchronous ready enable bit sets the Local Bus logic to wait for an external ready signal before closing the current cycle. 0 = Do not use, 1 = Use
Bits 13-11	CHT	Command Hold Time , and is determined as follows: Command hold time = (4+Vh) x T where Vh = Register value for the Hold time T = HCLK period
Bits 10-3	CAT	Command Active Time , and is determined as follows: Command active time = (2+Va) x T where Va = Register value for the Active time T = HCLK period.
Bits 2-0	CST	Command Setup Time , and is determined as follows: Command setup time = (4+Vs) x T where Vs = Register value for the Setup time T = HCLK period.



11.6.8. I/O TIMING TEMPLATE REGISTER 5

TIMEIO5 defines the timing template for accessing device in I/O Slot 5. The timing is programmed with reference to the host clock period as a time unit.

TIMEIO5

Access = see [Section 11.5](#).

Regoffset = 2Eh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSH	UARS	CHT			CAT								CST		
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bit 15	DSH	Discard the Setup and Hold parameters for read operations. 0 = Do not discard, 1 = discard
Bit 14	UARS	Use Asynchronous Ready Signal for command termination. The asynchronous ready enable bit sets the Local Bus logic to wait for an external ready signal before closing the current cycle. 0 = Do not use, 1 = Use
Bits 13-11	CHT	Command Hold Time , and is determined as follows: Command hold time = $(4+V_h) \times T$ where V_h = Register value for the Hold time T = HCLK period
Bits 10-3	CAT	Command Active Time , and is determined as follows: Command active time = $(2+V_a) \times T$ where V_a = Register value for the Active time T = HCLK period.
Bits 2-0	CST	Command Setup Time , and is determined as follows: Command setup time = $(4+V_s) \times T$ where V_s = Register value for the Setup time T = HCLK period.

11.6.9. I/O TIMING TEMPLATE REGISTER 6

TIMEIO6 defines the timing template for accessing device in I/O Slot 6. The timing is programmed with reference to the host clock period as a time unit.

TIMEIO6														Access = see Section 11.5 .			Regoffset = 30h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
DSH	UARS	CHT			CAT								CST						
Default value after reset = 0000h																			

Bit Number	Mnemonic	Description
Bit 15	DSH	Discard the Setup and Hold parameters for read operations. 0 = Do not discard, 1 = discard
Bit 14	UARS	Use Asynchronous Ready Signal for command termination. The asynchronous ready enable bit sets the Local Bus logic to wait for an external ready signal before closing the current cycle. 0 = Do not use, 1 = Use
Bits 13-11	CHT	Command Hold Time , and is determined as follows: Command hold time = (4+Vh) x T where Vh = Register value for the Hold time T = HCLK period
Bits 10-3	CAT	Command Active Time , and is determined as follows: Command active time = (2+Va) x T where Va = Register value for the Active time T = HCLK period.
Bits 2-0	CST	Command Setup Time , and is determined as follows: Command setup time = (4+Vs) x T where Vs = Register value for the Setup time T = HCLK period.

11.6.10. I/O TIMING TEMPLATE REGISTER 7

TIMEIO7 defines the timing template for accessing device in I/O Slot 7. The timing is programmed with reference to the host clock period as a time unit.

TIMEIO7

Access = see [Section 11.5](#).

Regoffset = 32h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSH	UARS	CHT			CAT								CST		
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bit 15	DSH	Discard the Setup and Hold parameters for read operations. 0 = Do not discard, 1 = discard
Bit 14	UARS	Use Asynchronous Ready Signal for command termination. The asynchronous ready enable bit sets the Local Bus logic to wait for an external ready signal before closing the current cycle. 0 = Do not use, 1 = Use
Bits 13-11	CHT	Command Hold Time , and is determined as follows: Command hold time = $(4+V_h) \times T$ where V_h = Register value for the Hold time T = HCLK period
Bits 10-3	CAT	Command Active Time , and is determined as follows: Command active time = $(2+V_a) \times T$ where V_a = Register value for the Active time T = HCLK period.
Bits 2-0	CST	Command Setup Time , and is determined as follows: Command setup time = $(4+V_s) \times T$ where V_s = Register value for the Setup time T = HCLK period.

11.7. LOCAL BUS CONTROL REGISTER

This 16-bit register controls the basic functionality of the local bus interface.

CONTROL

Access = see [Section 11.5](#).

Regoffset = 34h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv															
Default value after reset = 0001h															

Bit Number	Mnemonic	Description
Bits 15-12	Rsv	Reserved
11	OIOCS	OIOCS Selection Control; This enables the access of all 8 IOCS# using only 4 inputs. When this bit is set '1'; IOCS#[0] = IOCS#[0] & IOCS#[4]; IOCS#[1] = IOCS#[1] & IOCS#[5]; IOCS#[2] = IOCS#[2] & IOCS#[6]; IOCS#[3] = IOCS#[3] & IOCS#[7]; When set to '0' only IOCS#[3:0] are used.
Bit 10	ISE	Interrupt Support Enable 0 = Disable, 1 = Enable
Bit 9	EMCB0	Enable Mapping to segment C of Bank0² 0 = Disable, 1 = Enable
Bit 8	EMDB0	Enable Mapping to segment D of Bank0² 0 = Disable, 1 = Enable
Bit 7	EMEB0	Enable Mapping to segment E of Bank0² 0 = Disable, 1 = Enable
Bit 6		Splitting for 8-bit device 0 = use bs 8 signal generation 1 = use internal state machine logic
Bit 5		Splitting for 16-bit device 0 = use bs 16 signal generation 1 = use internal state machine logic
Bit 4	CEB1	Cache Enable for Bank1, 0 = Disable, 1 = Enable
Bit 3	CEB0	Cache Enable for Bank0, 0 = Disable, 1 = Enable
Bit 2	WEB1	Write Enable for Bank1, 0 = Disable, 1 = Enable
Bit 1	WEB0	Write Enable for Bank0, 0 = Disable, 1 = Enable
Bit 0	RMBAE	Real Mode Boot Access Enable 0 = Disable, 1 = Enable ¹

Notes:

- 1) Generation of Chip select for Bank0 at 0x000Fh:XXXX address.
- 2) Enable mapping to C,D,E segments means enabling Bank0 for address range 0x000Ch:XXXX, 0x000Dh:XXXX and 0x000Eh:XXXX respectively.

11.8. LOCAL BUS DEVICE WIDTH REGISTER

This is a 16-bit register whose individual bits are used to tell the Local Bus if an 8 or 16-bit peripheral is attached to one of the eight I/O and two MEM slots. Only ten LSB bits of the register are used.

WIDTH						Access = see Section 11.5 .						Regoffset = 36h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv						DWB1	DWB0	IOW7	IOW6	IOW5	IOW4	IOW3	IOW2	IOW1	IOW0
Default value after reset = 0300h															

Bit Number	Mnemonic	Description
Bits 15-10	Rsv	Reserved
Bit 9	DWB1	Data Width of Bank1 0 = 8-bit, 1 = 16-bit
Bit 8	DWB0	Data Width of Bank0 ¹ 0 = 8-bit, 1 = 16-bit
Bit 7	IOW7	I/O 7 Data Width 0 = 8-bit, 1 = 16-bit
Bit 6	IOW6	I/O 6 Data Width 0 = 8-bit, 1 = 16-bit
Bit 5	IOW5	I/O 5 Data Width 0 = 8-bit, 1 = 16-bit
Bit 4	IOW4	I/O 4 Data Width 0 = 8-bit, 1 = 16-bit
Bit 3	IOW3	I/O 3 Data Width 0 = 8-bit, 1 = 16-bit
Bit 2	IOW2	I/O 2 Data Width 0 = 8-bit, 1 = 16-bit
Bit 1	IOW1	I/O 1 Data Width 0 = 8-bit, 1 = 16-bit
Bit 0	IOW0	I/O 0 Data Width 0 = 8-bit, 1 = 16-bit

Note: This is also controlled by a strap option input to Local Bus:

Strap bit = 1: The data width of Bank0 is 16-bit

Strap bit = 0: The data width of Bank0 is 8-bit.



12.GPIO Interface

12.1. Introduction

The GPIO Interface provides a general purpose 8-bit I/O facility, using eight dedicated device pins. The block has the following features:

- Built-in debounce logic for each Input port.
- Each GPIO port is configurable for Rise or Fall edge interrupt generation.
- Strap recording on all 8 ports.

Each GPIO port can be configured as an input or an output simply by programming the associated port direction control register. All GPIO ports are configured as inputs at reset, which also latches the input levels into the Strap Registers. The input states of the ports are thus recorded automatically at reset, and this can be used as a strap register anywhere in the system.

All the GPIO ports can be configured to generate rise or fall edge interrupts, but this can only be done when the ports are configured as inputs. These interrupt states can be cleared from the Interrupt Edge Select Register by writing to the Clear Interrupt Register (Base+05h). In cascade mode it is also necessary to clear the interrupt of the Master GPIO.

12.2. GPIO Configuration

In order to configure the GPIO Interface, please refer to [Section 8.7.6. SOUTH BRIDGE MISCELLANEOUS REGISTER](#)

12.3. Register Description

The GPIO requires eight address lines to access all functions. The registers are explained in the following sections.

12.3.1. Port Direction Control Register (Base+00h):

This 8-bit register sets the direction, input or output, of each of the eight GPIO ports. After reset this register defaults to 0xFF, setting all ports to the input mode.

portDirCtrl		Access = h				Regoffset = 000h	
7	6	5	4	3	2	1	0
Port 7	Port 6	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0
Default value after reset = FFh							

Bit Number	Mnemonic	Description
Bits 7-0		0 = Output, 1= Input.



12.3.2. Read Port Control Register (Base+01h):

This is an 8-bit register which decides whether the data read at address Base+06H, for the ports which are configured as outputs, is the data at the GPIO Port or is the data from the read register at address Base+02h.

For all ports which are configured as Inputs, the read data at address Base+06h always returns the data at the GPIO Port.

For any port which is configured as an Output, the following holds true.

A ONE in any bit position returns the read data, at address Base+06h, from the Read Register, which is accessible at address Base+02h.

A ZERO in any bit position returns the read data, at address Base+06h, from the GPIO Port.

readPortCtrl				Access = h		Regoffset = 001h	
7	6	5	4	3	2	1	0
Port 7	Port 6	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0		1: Returns the read data, at address Base+06h, from the Read Register, which is accessible at address Base+02h. 0: Returns the read data, at address Base+06h, from the GPIO Port.

12.3.3. Read register (Base+02h):

This is an 8-bit register which is used to return the read data for the ports which are configured as Outputs, provided also that their corresponding bits in the Read Port Control Register are set to 1.

readReg		Access = h				Regoffset = 010h	
7	6	5	4	3	2	1	0
Port 7	Port 6	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0		Read data register for Ports configured as Outputs.



12.3.4. INTERRUPT UNMASK REGISTER (Base+03h):

This is an 8-bit register with each bit controlling the interrupt mask for the corresponding port. After reset this register is cleared, disabling all interrupt generation. Interrupts can be un-masked only for the ports which are configured as inputs. Clearing the mask for the ports which are configured as outputs has no effect.

intrUnMask

Access = h

Regoffset = 011h

7	6	5	4	3	2	1	0
Port 7	Port 6	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0		Write interrupt mask bits for selected input ports.

12.3.5. INTERRUPT EDGE REGISTER (Base+04h):

This is an 8-bit register. Each bit controls the trigger for interrupt generation for the corresponding ports. This has no effect if its corresponding interrupt is masked. After reset this register is cleared, which means that if the port's interrupt is unmasked, then the interrupt trigger condition is the rising edge on the input port. Again, this register is don't care for all ports which are configured as outputs.

intrEdgeSelect				Access = h		Regoffset = 100h	
7	6	5	4	3	2	1	0
Port 7	Port 6	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0		Write interrupt edge select bits for selected input ports 0: rising edge 1: falling edge



12.3.6. INTERRUPT CLEAR COMMAND(Base+05h):

This register has two functions depending on whether it is a read or a write access. If this is a write, then the Interrupt Clear Command (ICC) register is modified as described here. If this is a read, then the Interrupt Request Register (IRR) is returned. The IRR is located at the same address (Base+05h) and contains the status of the data ports that have generated an interrupt since the last write to the ICC.

Each bit value means: 0 = no interrupt, 1 = interrupt is pending on the selected port.

clearIntr		Access = h				Regoffset = 101h	
7	6	5	4	3	2	1	0
Port 7	Port 6	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0
Default value after reset = h							

Bit Number	Mnemonic	Description
Bits 7-0		Write 1 to clear all interrupts in write access and set a request in read access.

12.3.7. GPIO port register (Base+06h):

This is an 8-bit register which controls/reads the GPIO port. A write to this with any data will change the value on the ports which are configured as outputs. A write to this register has no effect on ports configured as inputs. Whereas a read to this address will return the value at the port for those ports which are configured as inputs. For those ports which are configured as outputs, a read will return either the output value OR will return the data from the read register, depending on the programming of the readPortCtrl register.

GPIOport			Access = h			Regoffset = 110h	
7	6	5	4	3	2	1	0
Port 7	Port 6	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0
Default value after reset = h							

Bit Number	Mnemonic	Description
Bits 7-0		Write output port data, read input port data



12.3.8. STRAP REGISTER (Base+07h):

This is an 8-bit register and is used to latch the value on the GPIO port at reset. This therefore becomes a strap register and can be used anywhere in the system. This register is read only; a write to this register has no effect.

strapReg

Access = h

Regoffset =111h

7	6	5	4	3	2	1	0
Port 7	Port 6	Port 5	Port 4	Port 3	Port 2	Port 1	Port 0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0		Read and latch GPIO port state at reset

13. UNIVERSAL SERIAL BUS

13.1. INTRODUCTION

The Universal Serial Bus (USB) is a general-purpose, high-speed, communications interface for connecting peripheral equipment to a PC. The main USB features are listed below:

- Using a special 4-wire cable, the bus can be expanded, using multi-port hubs, to link up to 127 devices to a PC.
- The USB supplies the power to the connected peripheral device, obviating the need for external power supplies. This is only true in the case of a device which is not self powered and does not require more than 500mA (normally, 1 device uses 100mA)
- The USB port connector is hot-pluggable, i.e. peripherals can be connected or disconnected with power applied to the PC.
- The Plug-and-Play concept is supported, where the PC recognises each peripheral device that is plugged in and loads the appropriate driver.
- Where a new peripheral device has no driver, and cannot run using a generic driver, the user is prompted for a driver to be loaded.
- Simple standard cabling requirements: no null modem cables, no handshaking lines, etc.
- Full speed devices can communicate with the PC at 12 Mbps; keyboards, mice, etc. can communicate at a lower 1.5 Mbps rate to reduce cost.

There are four data transfer types defined in USB. Each type is optimised to match the service requirements between the client software and the USB device. The four types are:

- 1) Isochronous Transfers: Periodic data transfers with a constant data rate. Data transfers are correlated in time between the sender and the receiver.
- 2) Control Transfers(Mandatory): Non-periodic data transfers used to communicate and control configuration/command/status type information between the client software and the USB device.
- 3) Interrupt Transfers: Small data transfers used to communicate information from the USB device to the client software. The Host Controller Driver polls the USB device by issuing tokens to the device at a periodic interval, sufficient for the requirements of the device.
- 4) Bulk Transfers: Non-periodic data transfers used to communicate large amounts of information between client software and the USB device. (Data Integrity Ensured)

The USB conforms to the *Open Host Controller Interface (OpenHCI) Specification, Release 1.1*, which provides a register-level description of a Host Controller for the Universal Serial Bus. Refer to this specification document for detailed information.

13.2. OPERATIONAL REGISTERS

The Host Controller (HC) operational registers are mapped into a non-cacheable portion of the system addressable space. These registers, which are used by the Host Controller Driver (HCD), are divided into four partitions:

- Control and Status
- Memory Pointer
- Frame Counter
- Root Hub

As reserved bits may be allocated in future releases, to ensure interoperability, the Host Controller Driver that does not use a reserved field should not assume that the reserved field contains 0. The Host Controller Driver should always preserve the value(s) of the reserved field. When a R/W register is modified, the Host Controller Driver should first read the register, modify the bits desired, then write the register with the reserved bits still containing the read value. Alternatively, the Host Controller Driver can maintain an in-memory copy of previously written values that can be modified and then written to the Host Controller register. When a write to set/clear register is written, bits written to reserved fields should be 0.

Table 13-1. Host Controller Operational Registers

Mnemonic	Offset	Description
Control and Status Partition:		
HcRevision	0	This 8-bit read-only field contains the BCD representation of the version of the HCI specification that is implemented by this HC. For example, a value of 11h corresponds to version 1.1. All of the HC implementations that are compliant with this specification will have a value of 10h.
HcControl	4	This register defines the operating modes for the Host Controller. Apart from HostControllerFunctionalState and RemoteWakeup-Connected , the fields in this register are modified only by the Host Controller Driver
HcCommandStatus	8	The <i>HcCommandStatus</i> register is used by the Host Controller to receive commands issued by the Host Controller Driver, as well as reflecting the current status of the Host Controller. To the Host Controller Driver, it appears to be a "write to set" register. The Host Controller must ensure that bits written as '1' become set in the register while bits written as '0' remain unchanged in the register. The Host Controller Driver may issue multiple distinct commands to the Host Controller without concern for corrupting previously issued commands. The Host Controller Driver has normal read access to all bits. The SchedulingOverrunCount field indicates the number of frames with which the Host Controller has detected the scheduling overrun error. This occurs when the Periodic list does not complete before EOF. When a scheduling overrun error is detected, the Host Controller increments the counter and sets the SchedulingOverrun field in the <i>HcInterruptStatus</i> register.



Mnemonic	Offset	Description
HcInterruptStatus	C	This register provides status on various events that cause hardware interrupts. When an event occurs, Host Controller sets the corresponding bit in this register. When a bit becomes set, a hardware interrupt is generated if the interrupt is enabled in the <i>HcInterruptEnable</i> register and the MasterInterruptEnable bit is set. The Host Controller Driver may clear specific bits in this register by writing '1' to bit positions to be cleared. The Host Controller Driver may not set any of these bits. The Host Controller will never clear the bit.
HcInterruptEnable	10	Each enable bit in the <i>HcInterruptEnable</i> register corresponds to an associated interrupt bit in the <i>HcInterruptStatus</i> register. The <i>HcInterruptEnable</i> register is used to control which events generate a hardware interrupt. When a bit is set in the <i>HcInterruptStatus</i> register AND the corresponding bit in the <i>HcInterruptEnable</i> register is set AND the MasterInterruptEnable bit is set, then a hardware interrupt is requested on the host bus. Writing a '1' to a bit in this register sets the corresponding bit, whereas writing a '0' to a bit in this register leaves the corresponding bit unchanged. On read, the current value of this register is returned.
HcInterruptDisable	14	Each disable bit in the <i>HcInterruptDisable</i> register corresponds to an associated interrupt bit in the <i>HcInterruptStatus</i> register. The <i>HcInterruptDisable</i> register is coupled with the <i>HcInterruptEnable</i> register. Thus, writing a '1' to a bit in this register clears the corresponding bit in the <i>HcInterruptEnable</i> register, whereas writing a '0' to a bit in this register leaves the corresponding bit in the <i>HcInterruptEnable</i> register unchanged. On read, the current value of the <i>HcInterruptEnable</i> register is returned.
Memory Pointer Partition:		
HcHCCA	18	The <i>HcHCCA</i> register contains the physical address of the Host Controller Communication Area. The Host Controller Driver determines the alignment restrictions by writing all 1s to <i>HcHCCA</i> and reading the content of <i>HcHCCA</i> . The alignment is evaluated by examining the number of zeroes in the lower order bits. The minimum alignment is 256 bytes; therefore, bits 0 through 7 must always return '0' when read. This area is used to hold the control structures and the Interrupt table that are accessed by both the Host Controller and the Host Controller Driver.
HcPeriodCurrentED	1C	The <i>HcPeriodCurrentED</i> register contains the physical address of the current Isochronous or Interrupt Endpoint descriptor.
HcControlHeadED	20	The <i>HcControlHeadED</i> register contains the physical address of the first Endpoint Descriptor of the Control list.
HcControlCurrentED	24	The <i>HcControlCurrentED</i> register contains the physical address of the current Endpoint Descriptor of the Control list.
HcBulkHeadED	28	The <i>HcBulkHeadED</i> register contains the physical address of the first Endpoint Descriptor of the Bulk list.

Mnemonic	Offset	Description
HcBulkCurrentED	2C	The <i>HcBulkCurrentED</i> register contains the physical address of the current endpoint of the Bulk list. As the Bulk list will be served in a round-robin fashion, the endpoints will be ordered according to their insertion to the list.
HcDoneHead	30	The <i>HcDoneHead</i> register contains the physical address of the last completed Transfer Descriptor that was added to the Done queue. In normal operation, the Host Controller Driver should not need to read this register as its content is periodically written to the HCCA.
Frame Counter Partition:		
HcFmInterval	34	The <i>HcFmInterval</i> register contains a 14-bit value which indicates the bit time interval in a Frame, (i.e., between two consecutive SOFs), and a 15-bit value indicating the Full Speed maximum packet size that the Host Controller may transmit or receive without causing scheduling overrun. The Host Controller Driver may carry out minor adjustment on the FrameInterval by writing a new value over the present one at each SOF. This provides the programmability necessary for the Host Controller to synchronise with an external clocking resource and to adjust any unknown local clock offset.
HcFmRemaining	38	The <i>HcFmRemaining</i> register is a 14-bit down counter showing the bit time remaining in the current Frame.
HcFmNumber	3C	The <i>HcFmNumber</i> register is a 16-bit counter. It provides a timing reference among events happening in the Host Controller and the Host Controller Driver. The Host Controller Driver may use the 16-bit value specified in this register and generate a 32-bit frame number without requiring frequent access to the register.
HcPeriodicStart	40	The <i>HcPeriodicStart</i> register has a 14-bit programmable value which determines when is the earliest time HC should start processing the periodic list.
HcLSThreshold	44	The <i>HcLSThreshold</i> register contains an 11-bit value used by the Host Controller to determine whether to commit to the transfer of a maximum of 8-byte LS packet before EOF. Neither the Host Controller nor the Host Controller Driver are allowed to change this value.
Root Hub Partition:		
HcRhDescriptorA	48	The <i>HcRhDescriptorA</i> register is the first register of two describing the characteristics of the Root Hub. Reset values are implementation-specific. The Descriptor length (11), Descriptor type (TBD), and hub controller current (0) fields of the hub Class Descriptor are emulated by the HCD. All other fields are located in the <i>HcRhDescriptorA</i> and <i>HcRhDescriptorB</i> registers.
HcRhDescriptorB	4C	The <i>HcRhDescriptorB</i> register is the second register of two describing the characteristics of the Root Hub. These fields are written during initialization to correspond with the system implementation. Reset values are implementation-specific.

Mnemonic	Offset	Description
HcRhStatus	50	The <i>HcRhStatus</i> register is divided into two parts. The lower word of a Dword represents the Hub Statusfield and the upper word represents the Hub StatusChange field. Reserved bits should always be written '0'.
HcRhPortStatus[1]	54	The <i>HcRhPortStatus</i> [1:NDP] register is used to control and report port events on a per-port basis. NumberDownstreamPorts represents the number of <i>HcRhPortStatus</i> registers that are implemented in hardware. The lower word is used to reflect the port status, whereas the upper word reflects the status change bits. Some status bits are implemented with special write behaviour (see below). If a transaction (token through handshake) is in progress when a write to change port status occurs, the resulting port status change must be postponed until the transaction completes. Reserved bits should always be written '0'.
...	...	
HcRhPortStatus[NDP]	54+4*N DP	

For a detailed Register description, refer to the *Open Host Controller Interface (OpenHCI) Specification, Release 1.1*.

13.3. PCI CONFIGURATION

13.3.1. PCI INTERFACE

This section describes the configuration registers used to interface with other system components in a PCI-based PC host. Only those bits relevant to the implementation of a USB Host Controller with PCI interface are described here. For the definition of the other bits/registers, refer to the PCI Specification, Revision 2.1.

The registers listed here are accessed for set-up during PCI initialization. They might also be accessed through special cycles during normal system runtime. *Header type 0* is the format for the device's configuration header region, the first 16 Dwords. They are also commonly called the *PCI configuration spaces* of a PCI device. For the USB Host Controller with PCI interface, the operational registers (i.e., *PCI nonconfiguration spaces*) are described in the Operational Registers section ([Section 13.2](#)). "Reset" issued to the Host Controller through its respective programming interface does not affect the contents of the *PCI configuration space* (contents of the operational registers of the Root Hub are also not affected). "Hardware reset" issued by the system logic in the PC host, during system power-up and "Hard reset", causes all of the on-chip registers of the Host Controller and the Root Hub to return their default values.

Note: The LATENCY_TIMER in the *PCI configuration spaces* defines the minimum amount of time that the Host Controller is permitted to retain ownership of the bus after it has acquired bus ownership and has initiated a subsequent transaction. It should be set to a value that reflects the nominal burst size of the underlying device, resulting in a good compromise between the utilization and efficiency of the PCI bus. In determining the value, it should be considered that the maximum size of packet transferred over the USB ranges from 8 bytes to 1023 bytes. A value of '16h' is recommended, as it will allow a total of 24 PCI clocks, sufficient for a burst transfer of 64-byte (assuming a target initial latency of 8 PCI clocks).

Like the other integrated peripherals, the USB Controller uses PCI Interrupt A.

—

13.3.2. PCI CONFIGURATION SPACES FOR OPENHCI-COMPLIANT USB HOST CONTROLLER

The following table provides a summary of the registers that are necessary for the USB Host Controller to be successfully configured in a PCI-based PC host.

Register	Offset	Description
Command	05 - 04	Provides coarse control over a device's ability to generate and respond to PCI cycles
CLASS_CODE	0B - 09	Identifies the generic function of the device
BAR_OHCI	13 - 10	Specifies the base address of a contiguous block in the main memory of the PC host, from which 4 KB of directly-mapped addressing spaces are reserved by OpenHCI for the operational registers of the Host Controller

For a detailed Register description, refer to the *Open Host Controller Interface (OpenHCI) Specification, Release 1.1*.



14. MAC ETHERNET INTERFACE (LAN)

14.1. INTRODUCTION

The MAC (Media Access Controller) Ethernet Interface is compliant with IEEE 802.3 and supports 10/100 Mb/s data transfer rates. A Media Independent Interface (MII) is used to talk to an external Physical layer (PHY).

14.2. FEATURES

- VLAN support.
- Supports both full-duplex/half-duplex operation.
- Support of CSMA/CD Protocol for half-duplex operation.
- Supports flow-control for full-duplex operation.
- Collision detection and auto re-transmission on collisions in half-duplex mode.
- Management support by using variety of counters.
- Preamble generation and removal.
- Automatic 32-bit CRC generation and checking.
- Options to insert PAD/CRC32 on transmit.
- Options for Automatic Pad stripping on the receive packets.
- Provides external and internal loop-back capability on the MII Interface.

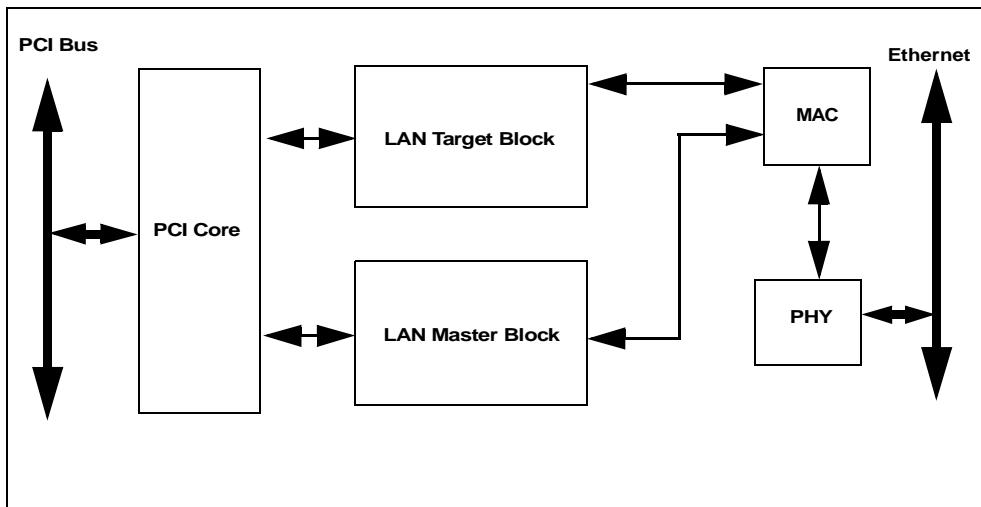


Figure 14-1. Block Diagram: MAC Ethernet Controller

14.3. FUNCTIONAL DESCRIPTION

The functional block diagram of the MAC Ethernet Controller given in [Figure 14-1](#) is explained briefly as follows:

The LAN (Local Area Network) glue logic consists of two main parts, namely:

LAN Target Block ([Section 14.3.1.](#))

LAN Master Block ([Section 14.3.2.](#))

14.3.1. LAN TARGET BLOCK

This comprises the Control & Status Registers (CSRs) and the state machine which communicate with and program the MAC Registers. These MAC registers are mapped to the same address space as that of the CSRs in the LAN Target block. The data transfer used for programming the MAC register set is a superset of the VSI Bus Protocol.

14.3.2. LAN MASTER BLOCK

This comprises the state machine which receives the data from the MAC block which is then buffered in the receive FIFO registers. The protocol for this DWord data transfer is a super set of VSI Bus Protocol. It also contains the Rx FIFO controller, a dual-port RAM buffer for data received from the MAC, and the state machine which transfers the data from the receive FIFO to the PCI core; this will, in turn, generate the PCI cycles on the PCI bus to transfer the data to memory.

This block also contains the state machine for the data from the PCI block (which initiates memory read cycles on the PCI bus and gets the data), together with buffers feeding the Transmit FIFO and a dual-port RAM to buffer the data to be transmitted to the MAC block. The protocol used for the transfer is a superset of VSI Bus Protocol.

14.3.3. MAC BLOCK

The Media Access Controller interfaces with the PHY (physical layer). It has a set of programmable (by the CPU) Control Registers, which sit in the same memory space as the CSRs in the LAN glue block.

The MAC Receiver obtains nibbles of data from the PHY, gathers them and then transfers them to the LAN glue block as DWords. The transfer protocol is a super set of the VSI Bus Protocol.

Conversely, the MAC Transmitter gathers DWords of data from the LAN glue block (using the same Protocol as the Receiver) and transmits them to the PHY as nibbles.

14.3.4. PHY BLOCK

This is the physical layer which transfers data to and from the Ethernet cable.

14.4. I/O SPACE REGISTER DESCRIPTION

The LAN Controller contains a set of on-chip operational registers which are mapped into a non-cacheable portion of the system addressable space. These registers are used by the Driver. According to the function of these registers, they are divided into four partitions, specifically for Control and Status, and Memory Pointers. All of the registers should be read and/or written as DWords.

Reserved bits may be allocated in future releases of this specification. To ensure interoperability, a Driver that does not use a reserved field should not assume that the reserved field contains 0. Furthermore, the Driver should always preserve the value(s) of the reserved field. When a R/W register is modified, the Host Controller Driver should first read the register, modify the bits desired, then write the register with the reserved bits still containing the read value. Alternatively, the Driver can maintain an in-memory copy of previously written values that can be modified and then written to the Controller register. When a write to set/clear register is written, bits written to reserved fields should be 0.

Table 14-1. Ethernet Control / Status Registers

Offset	Name	Size	Type	Section Reference
00h	Bus Mode Register	32 bit	RW	Section 14.5.1.
04h	Transmit Poll Demand Register	8 bit	RW	Section 14.5.2.
08h	Receive Poll Demand Register	8 bit	RW	Section 14.5.3.
0Ch	Rx Descriptor Ring Base Address Register	32 bit	RW	Section 14.5.4.
10h	Tx Descriptor Ring Base Address Register	32 bit	RW	Section 14.5.5.
14h	Status Register	32 bit	RO	Section 14.5.6.
1Ch	Interrupt Enable Register	32 bit	RW	Section 14.5.7.
2Ch	Current Tx Descriptor Pointer Register	32 bit	RW	Section 14.5.8.
30h	Current Rx Descriptor Pointer Register	32 bit	RW	Section 14.5.9.
18h	MAC Control Register	32 bit	RW	Section 14.5.10.
34h	MAC Address Low Register	32 bit	RW	Section 14.5.11.
38h	MAC Address Hi Register	32 bit	RW	Section 14.5.11.
3Ch	Multicast Address Low Register	32 bit	RW	Section 14.5.12.
40h	Multicast Address Hi Register	32 bit	RW	Section 14.5.12.
44h	MII Address Register	32 bit	RW	Section 14.5.13.
48h	MII Data Register	32 bit	RO	Section 14.5.14.
4Ch	Flow Control Register	32 bit	RW	Section 14.5.15.
50h	VLAN1 Tag Register	32 bit	RW	Section 14.5.16.
54	VLAN2 Tag Register	32 bit	RW	Section 14.5.17.

Table 14-2. Ethernet - PCI Configuration Registers (Function2)

Offset	Name	Size	Type	Section Reference
01	Vendor ID Register	16 bit	RO	Section 8.12.
03	Device ID Register	16 bit	RO	Section 8.12.
05	Command Register	16 bit	RW	Section 8.12.1.
06	Status Register	16 bit	RO	Section 8.12.2.
08	Revision ID Register	8 bit	RO	Section 8.12.3.
09	Program Interface	8 bit	RO	
0B	Class Code Register	24 bit	RO	Section 8.12.4.
0C	Cache Line Size Register	8 bit	RO	Section 8.12.5.
0D	Latency Timer Register	8 bit	RO	Section 8.12.6.
0E	Header Type Register	8 bit	RO	Section 8.12.7.
0F	BIST Register	8 bit	RO	Section 8.12.8.
10	Memory Base Address Register	32 bit	RO	Section 8.12.9.
14	I/O Base Address Register	32 bit	RO	Section 8.12.10.
3C	Interrupt Line Register	8 bit	RW	Section 8.12.11.
3D	Interrupt Pin Register	8 bit	RW	Section 8.12.12.

14.5. REGISTER BIT FIELD DESCRIPTIONS

14.5.1. BUS MODE REGISTER

BUS_MODE																Access = h		Regoffset = 0x00	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
Rsv												TAP			Rsv				
Default value after reset = 0000h																			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv		PBL						Rsv							SWR
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bits 31:20	Rsv	Reserved.
Bits 19:17	TAP	Transmit Automatic Polling When set and Tx is in a suspended state because a transmit buffer is unavailable, a transmit automatic poll demand is performed. <u>Bits 19:17 Polling Interval</u> 000 TAP Disabled 001 200 microseconds 010 800 microseconds Others 1600 microseconds
Bits 16:14	Rsv	Reserved.
Bits 13:8	PBL	Programmable Burst Length Indicates the maximum number of longwords to be transferred in one DMA transaction. If reset, the burst is limited only by the amount of data stored in the receive FIFO (at least 16 longwords), or by the amount of free space in the transmit FIFO (at least 16 longwords) before issuing a bus request. This can be programmed with permissible values of 0, 1, 2, 4, 8, 16 or 32.
Bit 7:1	Rsv	Reserved.
Bit 0	SWR	Software Reset Setting of this bit resets all internal hardware with the exception of the configuration area.



14.5.2. TRANSMIT POLL DEMAND REGISTER

TRANSMIT_POLL_DEMAND Access = h Regoffset = 0x04

7	6	5	4	3	2	1	0
NA							DMD

Bit Number	Mnemonic	Description
Bits 7:1	NA	Not Applicable.
Bit 0	DMD	Transmit Poll Demand When written with '1', the Transmitter is triggered to resume from the suspended state.



14.5.3. RECEIVE POLL DEMAND REGISTER

RECEIVE_POLL_DEMAND				Access = h			Regoffset = 0x08	
7	6	5	4	3	2	1	0	
NA							DMD	

Bit Number	Mnemonic	Description
Bits 7:1	NA	Not Applicable.
Bit 0	DMD	Receive Poll Demand When written with '1', the Receiver is triggered to resume from the suspended state.



14.5.4. RECEIVE DESCRIPTOR RING BASE ADDRESS

Rx_DESCR_BASE ADD Access = h Regoffset = 0x0Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Start Address of Rx Descriptor List															
Default value after reset = 0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start Address of Rx Descriptor List														LA	
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bits 31:2	SA	Start Address This is the pointer to the Start Address location of the Receive Descriptors.
Bits 0:1	LA	Longword Align This is always DWord aligned, so the least significant two bits are wired to zero.



14.5.5. TRANSMIT DESCRIPTOR RING BASE ADDRESS

Tx_DESCR_BASE ADD																Access = h				Regoffset = 0x10h			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
Start Address of Tx Descriptor List																							
Default value after reset = 0000h																							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Start Address of Tx Descriptor List									
														LA		Default value after reset = 0000h									

Bit Number	Mnemonic	Description
Bits 31:2	SA	Start Address This is the pointer to the Start Address location of the Transmit Descriptors.
Bits 0:1	LA	Longword Align This is always DWord aligned, so the least significant two bits are wired to zero.



14.5.6. STATUS REGISTER

STATUS

Access = h

Regoffset = 0x14h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv									TS			RS			NIS
Default value after reset = 0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIS	Rsv	FBE	Rsv			RWT	RPS	RU	RI	UNF	Rsv		TU	TPS	TI
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bits 31:23	Rsv	Reserved.
Bits 22:20	TS	Transmit Process State: 000: Stopped - Reset command issued. Tx Enable reset. 001: Running - Fetching transmit descriptor. 010: Running - Waiting for end of transmission. 011: Running - Reading buffer from memory and queuing the data into the FIFO. 100: Reserved. 101: Running - Setup packet. 110: Suspended - Transmit FIFO underflow, or an unavailable Tx descriptor. 111: Running - Closing transmit descriptor.
Bits 19:17	RS	Receive Process State: 000: Stopped - Reset command issued. 001: Running - Fetching receive descriptor. 010: Running - Checking for end of receive packet before prefetch of next descriptor. 011: Running - Waiting for receive packet. 100: Suspended - Unavailable receive buffer. 101: Running - Closing receive descriptor. 110: Running - Flushing the current frame from the receive FIFO as Receive buffer unavailable. 111: Running - Queuing the receive frame from the receive FIFO into the receive buffer.

Bit 16	NIS	Normal Interrupt Summary: The Normal Interrupt Summary bit is the logical <i>OR</i> of : [0] - Transmit Interrupt (TI) [2] - Transmit Buffer Unavailable (TU) [6] - Receive Interrupt (RI) Only unmasked bits affect the NIS bit.
Bit 15	AIS	Abnormal Interrupt Summary: This is the logical <i>OR</i> of: [1] - Transmit Process Stopped (RPS) [5] - Transmit Underflow (UNF) [7] - Receive Buffer Unavailable (RU) [8] - Receive Process Stopped (RPS) [13] - Fatal Bus Error (FBE).
Bit 14	Rsv	Reserved.
Bit 13	FBE	Fatal Bus Error: Indicates that a bus error has occurred. When this bit is set, all bus access operations are suspended.
Bits 12:10	Rsv	Reserved.
Bit 9	RWT	Receive Watchdog Timeout: This bit reflects the line status and indicates that the receive watchdog timer has expired while another node is still active on the network. In case of overflow, the long packets may not be received.
Bit 8	RPS	Receive Process Stopped: Asserted when the receive process enters the stopped state.
Bit 7	RU	Receive Buffer Unavailable: Indicates that the next descriptor in the receive list is owned by the host and cannot be acquired by the Receiver block. The reception process is suspended. To resume processing receive descriptors, the host should change the ownership of the descriptor and issue a receive poll demand command. If no receive poll demand is issued, the reception process resumes when the next recognized incoming frame is received.
Bit 6	RI	Receive Interrupt: Indicates the completion of a frame reception. Specific frame status information has been posted in the descriptor. The reception process remains in the running state.
Bit 5	UNF	Transmit Underflow: Indicates that the transmit FIFO had an undreflow condition during the packet transmission. The transmit process is placed in the suspended state and underflow error is set in the Tx descriptor status field.
Bits 4:3	Rsv	Reserved.

Bit 2	TU	Transmit Buffer Unavailable: Indicates that the next descriptor on the transmit list is owned by the host and cannot be acquired by the Tx block. The transmission process is suspended. To resume processing transmit descriptors, the host should change the ownership bit of the descriptor and then issue a transmit poll demand command, unless the transmit automatic polling bit in Bus mode Register is enabled.
Bit 1	TPS	Transmit Process Stopped: Set when the transmit process enters the stopped state.
Bit 0	TI	Transmit Interrupt: Indicates that a frame transmission was completed and TDES1<31> is set in the first descriptor of the frame.

MAC ETHERNET INTERFACE (LAN)

14.5.7. INTERRUPT ENABLE REGISTER

This register enables the interrupts reported by the Status Register. Setting a bit enables the corresponding interrupt. After a hardware or software reset, all interrupts are disabled.

INTERRUPT_ENABLE Access = h Regoffset = 0x1Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															NIE
Default value after reset = F3FEh															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIE	Rsv	FBE	Rsv				RSE	RUE	RIE	UNE	Rsv		TUE	TSE	TIE
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bits 31:17	Rsv	Reserved.
Bit 16	NIE	Normal Interrupt Summary Enable: When set, normal interrupt is enabled. Since normal interrupt is a logical OR of many interrupt bits, the individual enabling of all those interrupts is required.
Bit 15	AIE	Abnormal Interrupt Summary Enable: When set, abnormal interrupt is enabled. Since abnormal interrupt is a logical OR of many interrupt bits, the individual enabling of all those interrupts is required.
Bit 14	Rsv	Reserved.
Bit 13	FBE	Fatal Bus Error Enable: When this bit and the Abnormal Interrupt summary Enable bit [15] are set, the fatal bus error interrupt is enabled.
Bits 12:9	Rsv	Reserved.
Bit 8	RSE	Receive Stopped Enable: When this bit and the Abnormal Interrupt summary Enable bit [15] are set, the receive stopped interrupt is enabled.
Bit 7	RUE	Receive Buffer Unavailable Enable: When this bit and the Abnormal Interrupt summary Enable bit [15] are set, the receive buffer unavailable interrupt is enabled.
Bit 6	RIE	Receive Interrupt Enable: When this bit and the Normal Interrupt summary Enable bit [16] are set, the receive interrupt is enabled.



Bit 5	UNE	Underflow Interrupt Enable: When this bit and the Abnormal Interrupt summary Enable bit [15] are set, the transmit underflow interrupt is enabled.
Bit3 4:3	Rsv	Reserved.
Bit 2	TUE	Transmit Buffer Unavailable Enable: When this bit and the Normal Interrupt summary Enable bit [16] are set, the transmit buffer unavailable interrupt is enabled.
Bit 1	TSE	Transmit Stopped Enable: When this bit and the Abnormal Interrupt summary Enable bit [15] are set, the transmit process stopped interrupt is enabled.
Bit 0	TIE	Transmit Interrupt Enable: When this bit and the Normal Interrupt summary Enable bit [16] are set, the transmit interrupt is enabled.

RWE - Receive Watchdog Timeout Enable - Bit 9, 10, 11 or 12? - if used at all?

When this bit and the Abnormal Interrupt summary Enable bit [15] are set, the receive watchdog timeout interrupt is enabled.

14.5.8. CURRENT TX DESCRIPTOR POINTER

This returns the pointer to the Tx descriptor that is currently being processed.

CURRENT Tx DP																Access = h		Regoffset = 0x2Ch			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16						
Current Transmit Descriptor Pointer																					
Default value after reset = 0000h																					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Current Transmit Descriptor Pointer																			
Default value after reset = 0000h																			



14.5.9. CURRENT RX DESCRIPTOR POINTER

This returns the pointer to the Rx descriptor that is currently being processed .

CURRENT Rx DP															
Access = h															
Regoffset = 0x30h															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Current Receive Descriptor Pointer															
Default value after reset = 0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Current Receive Descriptor Pointer															
Default value after reset = 0000h															



MAC ETHERNET INTERFACE (LAN)

14.5.10. MAC CONTROL REGISTER

The MAC Control Register establishes the receive and transmit operating modes and controls for address filtering and packet filtering.

MAC_CONTROL																Access = h	Regoffset = 0x18h
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
RA	BLE	Rsv	HBD	PS	SF	TC		DRO	OM		FD	PM	PR	IF	PB		
Default value after reset = 0000h																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HO	Rsv	HP	LCC	DBF	DRTY	Rsv	ASTP	BOLMT		DC	Rsv	TE	RE	DTE	Rsv
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bit 31	RA	Receive All: When set, all incoming packets will be received, regardless of the destination address. The address match is checked and is reported in Transmit Status (default: ' 0').
Bit 30	BLE	Indian Mode: When set, the MAC operates in the Big Endian mode. When reset, the MAC operates in the Little Endian mode. The Endian mode is for data buffers only (default: ' 0').
Bit 29	Rsv	Reserved.
Bit 28	HBD	Heart Beat Disable: When set, the heartbeat signal quality (SQE) generator function is disabled. This bit should be set in the MII Mode (default: ' 0').
Bit 27	PS	Port Select: When set, the MII/RMII port is selected, when reset, the SRL(ENDEC) port is selected for transmit/ receive operations on the Ethernet side.
Bit 26	SF	Store and Forward: When set, transmission starts when a full packet resides in the FIFO. When this occurs, the threshold values specified by Threshold Control Bits (25,24 of this MAC Control Reg) are ignored. This bit can be changed only when the transmit process is in the stopped state, otherwise, this bit will not be sampled.



Bits 25:24	TC	Threshold Control: Controls the selected threshold level for the transmit FIFO. The threshold value has a direct impact on the bus arbitration scheme. Transmission starts when the frame size within the transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. This bit can be changed only when the transmit process is in the stopped state, otherwise, this bit will not be sampled.
Bit 23	DRO	Disable Receive Own: When set, the MAC disables the reception of frames when the TXEN is asserted. The MAC will ignore any loop-backed receive packets. When reset, the MAC receives all the packets that are given by the PHY. This bit should be reset when the Full Duplex Mode bit is set or the Operating Mode is set to other than 'Normal Mode'.
Bits 22:21	OM	LoopBack Operating Mode: Selects the loopback operation modes for the MAC.
Bit 20	FD	Full Duplex Mode: When Set, the MAC operates in a full-duplex mode where it can transmit and receive simultaneously. While in full-duplex mode: heart-beat check is disabled, heartbeat fail status should be ignored, and internal loop-back is not allowed (default: '0').
Bit 19	PM	Pass All Multicast: When set, indicates that all the incoming frames with a multicast destination address (first bit in the destination address field is '1' are received. Incoming frames with physical address destinations are filtered only if the address matches with the MAC Address (default: '0').
Bit 18	PR	Promiscuous Mode: When set, indicates that any incoming valid frame is received regardless of its destination address (default: '1').
Bit 17	IF	Inverse Filtering: When set, Address Check block operates in the inverse filtering mode. This is valid only during perfect filtering mode (default: '0').
Bit 16	PB	Pass Bad Frames: When set, all incoming frames that passed the address filtering are received, including runt frames, collided frames, or truncated frames caused by Buffer underflow (default: '0').
Bit 15	HO	Hash Only Filtering Mode: When set, the Address Check block operates in the imperfect address filtering mode both for physical and multicast addresses.
Bit 14	Rsv	Reserved.

Bit 13	HP	Hash/Perfect Filtering Mode: When reset, the Address Check block does a perfect address filter of incoming frames according to the address specified in the MAC Address register. When set, the Address Check block does imperfect address filtering of multicast incoming frames according to the hash table specified in the multicast Hash Table Register. If the Hash Only(HO) is set, then physical addresses are imperfect filtered too. If Hash Only bit(HO) is reset, then physical addresses are perfect address filtered according to the MAC Address Register (default: '0').
Bit 12	LCC	Late Collision Control: When set, enables the re-transmission of the collided frame even after the collision period (late collision). When reset, the MAC110 Core disables the frame transmission on a late collision. In any case the Late Collision Status is appropriately updated in the Transmit Packet Status (default: '0').
Bit 11	DBF	Disable Broadcast Frames: When set, disables the reception of broadcast frames. When reset, forwards all the broadcast frames to the memory (default: '0').
Bit 10	DRTY	Disable Retry: When set, the MAC will attempt only one transmission. When a collision is seen on the bus the MAC will ignore the current frame, goes to the next frame. When reset, the MAC will attempt 16 transmissions.
Bit 9	Rsv	Reserved.
Bit 8	ASTP	Automatic Pad Stripping: When set, the MAC will strip the pad field on all the incoming frames, if the length field is less than 46 bytes. The FCS field will also be stripped, since it is computed at the transmitting station based on the data and pad field characters, and will be invalid for a receive frame that has had the pad characters stripped. Receive frames which have a length field of 46 bytes or greater will be passed to the host unmodified (FCS is not stripped). When reset, the MAC will pass all the incoming frames to the host unmodified (default: '0').
Bits 7:6	BOLMT	Backoff Limit: The Backoff limit determines the integer number of slot times the MAC waits before rescheduling a transmission attempt (during retries after a collision).

Bit 5	DC	Deferral Check: When set, the deferral check is enabled in the MAC. The MAC will abort the transmission attempt if it has deferred for more than 24,288 bit times. Deferring starts when the transmitter is ready to transmit, but is prevented from doing so because CRS is active. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of backoff, the deferral timer resets to 0 and restarts. When reset, the deferral check is disabled in the MAC and the MAC defers indefinitely (default: '0').
Bit 4	Rsv	Reserved.
Bit 3	TE	Transmitter Enable: When set, the MAC transmitter is enabled and it will transmit frames from the buffer on to the cable. When reset, the MAC transmitter is disabled and will not transmit any frames (default: '0').
Bit 2	RE	Receiver Enable: When set, the MAC receiver is enabled and will receive frames from the MII interface. When reset, the MAC receiver is disabled and will not receive any frames from the MII interface (default: '0').
Bit 1	DTE	DMA Transmitter Enable: This is exclusively for DMA. When set, the DMA transmitter is enabled and it will receive frames from the PCI and store them in the buffer (Tx FIFO). When this bit is reset, the DMA transmitter will finish processing the current frame and then stop. In other words, the DMA samples this bit only when a frame processing is over and before starting the next frame.
Bit 0	Rsv	Reserved.

MAC ETHERNET INTERFACE (LAN)

14.5.11. MAC ADDRESS HI AND LO REGISTERS

The MAC Address Hi Register contains the upper 16 bits of the physical address of the MAC. The MAC Address Lo Register contains the lower 32 bits of the physical address of the MAC.

MAC_ADDRESS_HI																Access = h		Regoffset = 0x38h			
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48						
Reserved																					
Default value after reset = FFFFh																					

47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32				
PADR-HI																			
Default value after reset = FFFFh																			

MAC_ADDRESS_LO																Access = h		Regoffset = 0x34h			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16						
PADR-LO																					
Default value after reset = FFFFh																					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
PADR-LO																			
Default value after reset = FFFFh																			

Bit Number	Mnemonic	Description
Bits 63:48	Rsv	Reserved



Bits 47:32	PADR-HI	This field contains the upper 16 bits of the Physical Address of the MAC device. The contents of this field can be updated by the host after the initialization process is completed.
Bits 31:0	PADR-LO	This field contains the lower 32 bits of the Physical Address of the MAC device. The contents of this field can be updated by the host after the initialization process is completed.



14.5.12. MULTICAST ADDRESS HI/LO REGISTERS

The Multi Cast Hash Table Hi Register contains the higher 32 bits of the hash table and the Multi Cast Hash Table Low Register contains the lower 32 bits of the hash table.

MULTICAST_ADDRESS_HI Access = h Regoffset = 0x40h

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
MA_HI															
Default value after reset = 0000h															

47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
MA_HI															
Default value after reset = 0000h															

MULTICAST_ADDRESS_LO Access = h Regoffset = 0x3Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA-LO															
Default value after reset = 0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MA-LO															
Default value after reset = 0000h															

The 64-bit multicast table is used for group address filtering. For hash filtering, the contents of the destination address in the incoming frame is passed through the CRC logic and the upper six bits of the CRC register are used to index the contents of the Hash table. The most significant bit determines the register to be used (Hi/Low), while the other five bits determine the bit within the register. A value of '00000' selects bit 0 of the selected register and a value of '11111' selects bit 31 of the selected register.

If the corresponding bit is '1', then the multicast frame is accepted, else it is rejected. If the Pass All Multicast is set, then all multi-cast frames are accepted regardless of the multi-cast hash values.



14.5.13. MII ADDRESS REGISTER

The MII Address Register is used to control the Management cycles to the External PHY Controller chip.

MI1_ADDRESS																Access = h		Regoffset = 0x44h			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Rsv					
Default value after reset = 0000h																					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY Add					MII Reg					Rsv				MIIW	MIIB
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bits 31-16	Rsv	Reserved.
Bits 15:12	PHY Add	PHY Address: These bits tell which of the 32 possible PHY devices are being accessed (default: 00000B).
Bits 10:6	MII Reg	MII Register: These bits select the desired MII register in the selected PHY device (default: 00000B).
Bits 5:2	Rsv	Reserved.
Bit 1	MIIW	MII Write: Setting this bit tells the PHY that this will be a write operation using the MII data register. If this bit is not set, this will be a read operation, placing the data in the MII data register (default: '0').
Bit 0	MIIB	MII Busy: This bit should read a logic 0 before writing to the MII address and MII data registers. This bit must also be set to 0 during write to the MII address register. During a MII register access, this bit will be set to signify that a read or write access is in progress. The MII data register should be kept valid until this bit is cleared by the MAC during a PHY write operation. The MII data register is invalid until this bit is cleared by the MAC during a PHY read operation. The MII address register should not be written to until this bit is cleared (default: '1').



14.5.14. MII DATA REGISTER

This register contains the data to be written to the PHY register specified in the MII Address Register, or it contains the read data from the PHY register, whose address is specified in the MII address register.

MIIData															
Access = h															
Regoffset = 0x48h															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = 0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MII Data															
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bits 31-16	Rsv	Reserved.
Bits 15:0	MII Data	This contains the 16-bit value read from the PHY after an MII read operation or the 16-bit data value to be written to the PHY before an MII write operation.



14.5.15. FLOW CONTROL REGISTER

This register is used to control the generation and reception of the Control (PAUSE Command) frames by the MAC Flow control block. A write to register with busy bit set to '1' triggers the Flow Control block to generate a Control frame. The fields of the control frame are selected as specified in the 802.3x specification and PauseTime value from this register is used in the "Pause Time" field of the control frame. The Busy bit is set until the control frame is transferred onto the cable. The Host has to make sure that the Busy bit is cleared before writing the register. The Pass Control Frames bit indicates to the MAC whether to pass the control frame to the Host or not and the Flow Control Enable bit enables the receive portion of the Flow Control block.

FLOW_CONTROL

Access = h

Regoffset = 0x4Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PT															
Default value after reset = 0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv													PCF	FCE	FCB
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bits 31-16	PT	Pause Time: This field contains the value that is to be used in the PAUSE TIME field in the control frame.
Bits 15:3	Rsv	Reserved.
Bit 2	PCF	Pass Control Frames: When set, the control frames are passed to the Host. The MAC will decode the control frame (PAUSE), and disable the transmitter for the specified amount of time. The Received Pause Command CSR5[27] bit is set and Transmitter Pause Mode CSR5[3] indicates the current state of the MAC Transmitter. When reset, the MAC will decode the control frames but will not pass the frames to the Host. The Received Pause Command CSR5[27] bit will be set and the Transmitter Pause Mode CSR5[3] gives the current status of the Transmitter.

Bit 1	FCE	Flow Control Enable: When set, the MAC is enabled for flow control operation and it will decode all the incoming frames for control frames. If the MAC receives a valid control frame (PAUSE command), it will disable the transmitter for the specified time. When reset, the flow control operation in the MAC is disabled and the MAC does not decode the frames for control frames.
Bit 0	FCB	Flow Control Busy: This bit should read a logic 0 before writing to the Flow Control register. To initiate a PAUSE control frame the host must set this bit to '1'. During a transfer of Control Frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion PAUSE control frame transmission, the MAC will reset to '0'. The Flow Control register should not be written to until this bit is cleared.



14.5.16. VLAN1 TAG REGISTER

This register contains the Length/Type field to identify the VLAN1 frames. The MAC compares the 13th and 14th bytes of the incoming frame (Length/Type) field and if a non-zero match occurs, it sets the VLAN1 tag to the received frame. The legal length of the frame is increased from 1518 bytes to 1522 bytes.

VLAN1_TAG																Access = h		Regoffset = 0x50h	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
																Rsv			
Default value after reset = 0000h																			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
VLAN1																			
Default value after reset = 0000h																			

Bit Number	Mnemonic	Description
Bits 31-16	Rsv	Reserved.
Bits 15:0	VLAN1	VLAN1 Tag Identifier: This contains the Length/Type field to identify the VLAN1 frames. This is used to compare with the 13th and 14th bytes of the incoming frames for VLAN1 frames.



14.5.17. VLAN2 TAG REGISTER

This register contains the Length/Type field to identify the VLAN2 frames. The MAC compares the 13th and 14th bytes of the incoming frame (Length/Type) field and if a non-zero match occurs, it sets the VLAN2 tag to the received frame. The legal length of the frame is increased from 1518 bytes to 1538 bytes.

VLAN2_TAG																Access = h	Regoffset = 0x54h
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Rsv																	
Default value after reset = 0000h																	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
VLAN2																	
Default value after reset = 0000h																	

Bit Number	Mnemonic	Description
Bits 31-16	Rsv	Reserved.
Bits 15:0	Rsv	VLAN2 Tag Identifier: This contains the Length/Type field to identify the VLAN2 frames. This is used to compare with the 13th and 14th bytes of the incoming frames for VLAN2 frames.

14.6. HOST COMMUNICATION

This describes descriptor lists and data buffers, collectively called the host communication area, which manages the actions and status related to buffer management.

Note: All reserved bits should be written by the driver as zero.

The hardware and the driver communicate through two data structures:

- (1) Control and status registers (CSRs), described above.
- (2) Descriptor lists and data buffers, listed below and described in the following paragraphs.

Offset	Name	Size	Type	Section Reference
	Receive Descriptor 0	32 bit	RW	Section 14.7.1.1.
	Receive Descriptor 1	32 bit	RW	Section 14.7.1.2.
	Receive Descriptor 2	32 bit	RW	Section 14.7.1.3.
	Receive Descriptor 3	32 bit	RW	Section 14.7.1.4.
	Transmit Descriptor 0	32 bit	RW	Section 14.7.2.1.
	Transmit Descriptor 1	32 bit	RW	Section 14.7.2.2.
	Transmit Descriptor 2	32 bit	RW	Section 14.7.2.3.
	Transmit Descriptor 3	32 bit	RW	Section 14.7.2.4.

14.7. DESCRIPTOR LISTS AND DATA BUFFERS

The hardware transfers received data frames to the receive buffers in host memory and transmits data from the transmit buffers in host memory. Descriptors that reside in the host memory act as pointers to these buffers. There are two descriptor lists, one for receive and one for transmit. The base address of each list is written into Rx Descriptor Ring Base Address and Tx Descriptor Ring Base Address Registers, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both the receive and transmit descriptors RDES1<24> and TDES1<24>.

The descriptor lists reside in the host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, and not contiguous in memory. A data buffer consists of either an entire frame or part of a frame, but it cannot exceed a single frame. Buffers contain only data; buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. Data chaining can be enabled or disabled. Data buffers reside in host physical memory space.

14.7.1. RECEIVE DESCRIPTORS (RDES)

Descriptors and receive buffers addresses must be longword aligned. Providing two buffers, two byte-count buffers, and two address pointers in each descriptor enables the adapter port to be compatible with various types of memory-management schemes.



MAC ETHERNET INTERFACE (LAN)

14.7.1.1. Receive Descriptor 0

RDES0		Access = h										Regoffset =				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
OWN	FF	FL														
Default value after reset = h																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	DE	Rsv		RF	MF	FS	LS	TL	CS	FT	RW	RE	DB	CE	ZERO
Default value after reset = h															

Bit Number	Mnemonic	Description
Bit 31	OWN	Own Bit When set, indicates that the descriptor is owned by the 21143. When reset, indicates that the descriptor is owned by the host. The 21143 clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
Bit 30	FF	Filtering Fail When set, indicates that the frame failed the address recognition filtering. This bit can be set only when receive all (CSR6<30>) is set. Otherwise, this bit is reset. This bit is valid only when last descriptor (RDES0<8>) is set and when the received frame is 64 bytes or longer.
Bits 29 -16	FL	Frame Length Indicates the length, in bytes, of the received frame that was transferred into host memory, including the cyclic redundancy check (CRC). Normally, this is also the length in bytes of the frame received from the network. In the case of receive timeout, the length of the frame on the network is longer. This field is valid only when last descriptor (RDES0<8>) is set and descriptor error (RDES0<14>) is reset.
Bit 15	ES	Error Summary Indicates the logical OR of the following RDES0 bits: RDES0<1> CRC error RDES0<6> Collision seen RDES0<7> Frame too long RDES0<11> Runt frame RDES0<14> Descriptor error This bit is valid only when last descriptor (RDES0<8>) is set.



Bit 14	DE	Descriptor Error When set, indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the 21143 does not own the next descriptor. The frame is truncated. This bit is valid only when last descriptor (RDES0<8>) is set.
Bits 13 - 12	Rsv	Reserved
Bit 11	RF	Runt Frame When set, indicates that this frame was damaged by a collision or premature termination before the collision window had passed. Runt frames are passed on to the host only if the pass bad frames bit (CSR6<3>) is set. This bit is valid only when last descriptor (RDES0<8>) is set.
Bit 10	MF	Multicast Frame When set, indicates that this frame has a multicast address. This bit is valid only when last descriptor (RDES0<8>) is set.
Bit 9	FS	First Descriptor When set, indicates that this descriptor contains the first buffer of a frame. If the buffer size of the first buffer is 0, the second buffer contains the beginning of the frame. If the buffer size of the second buffer is also 0, the second descriptor contains the beginning of the frame.
Bit 8	LS	Last Descriptor When set, indicates that the buffers pointed to by this descriptor are the last buffers of the frame.
Bit 7	TL	Frame Too Long When set, indicates that the frame length exceeds the maximum Ethernet-specified size of 1518 bytes. This bit is valid only when last descriptor (RDES0<8>) is set. Note: Frame too long is only a frame length indication and does not cause any frame truncation.
Bit 6	CS	Collision Seen When set, indicates that the frame was damaged by a collision that occurred after the 64 bytes following the start frame delimiter (SFD). This is a late collision. This bit is valid only when last descriptor (RDES0<8>) is set.
Bit 5	FT	Frame Type When set, indicates that the frame is an Ethernet-type frame (frame length field is greater than 1500 bytes). When clear, indicates that the frame is an IEEE 802.3 frame. This bit is not valid for runt frames of less than 14 bytes. This bit is valid only when last descriptor (RDES0<8>) is set.
Bit 4	RW	Receive Watchdog Timeout - Reserved
Bit 3	RE	Report on MII Error When set, indicates that a receive error in the physical layer was reported during the frame reception. This bit is valid only if the packet was received on the MII/SYM port and when last descriptor (RDES0<8>) is set.

Bit 2	DB	Dribbling Bit When set, indicates that the frame contained a non-integer multiple of 8 bits. This error is reported only if the number of dribbling bits in the last byte is 4 in MII operating mode, or at least 3 in 10-Mb/s serial operating mode. This bit is not valid if collision seen (RDES0<6>) is set. If set, and the CRC error (RDES0<1>) is reset, then the packet is valid. This bit is valid only when last descriptor (RDES0<8>) is set.
Bit 1	CE	CRC Error When set, indicates that a cyclic redundancy check (CRC) error occurred on the received frame. This bit is also set when the mii_err pin is asserted during the reception of a receive packet even though the CRC may be correct. This bit is not valid if one of the following conditions exist: The received frame is a runt frame A collision occurred while the packet was being received A watchdog timeout occurred while the packet was being received This bit is valid only when last descriptor (RDES0<8>) is set.
Bit 0	ZERO	This bit is always zero for a packet with legal length.



14.7.1.2. Receive Descriptor 1

RDES1

Access = h

Regoffset =

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv						RER	RCH	Rsv			RSB2				
Default value after reset = h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSB2					RSB1										
Default value after reset = h															

Bit Number	Mnemonic	Description
Bits 31 - 26	Rsv	Reserved
Bit 25	RER	Receive End of Ring When set, indicates that the descriptor list reached its final descriptor. The 21143 returns to the base address of the list , creating a descriptor ring.
Bit 24	RCH	Second Address Chained When set, indicates that the second address in the descriptor is the next descriptor address, rather than the second buffer address. RDES1<25> takes precedence over RDES1<24>.
Bits 23 - 22	Rsv	Reserved
Bits 21:11	RSB2	Buffer 2 Size Indicates the size, in bytes, of the second data buffer. If this field is 0, the 21143 ignores this buffer and fetches the next descriptor. The buffer size must be a multiple of 4. This field is not valid if RDES1<24> is set.
Bits 10:0	RBS1	Buffer 1 Size Indicates the size, in bytes, of the first data buffer. If this field is 0, the 21143 ignores this buffer and uses buffer 2. The buffer size must be a multiple of 4.

14.7.1.3. Receive Descriptor 2

RDES2																Access = h						Regoffset =																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																								
																BA1																							
Default value after reset = h																																							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BA1			
Default value after reset = h																			

Bit Number	Mnemonic	Description
Bits 31:0	BA1	Buffer Address 1 Indicates physical address of buffer 1. The buffer must be longword aligned (RDES2<1:0> = 00).



14.7.1.4. Receive Descriptor 3

RDES3

Access = h

Regoffset =

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA2															
Default value after reset = h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA2															
Default value after reset = h															

Bit Number	Mnemonic	Description
Bits 31:0	BA2	Buffer Address 1 Indicates physical address of buffer 2. The buffer must be longword aligned (RDES3<1:0> = 00).



14.7.2. TRANSMIT DESCRIPTORS

Descriptor addresses must be longword aligned. Providing two buffers, two byte-count buffers, and two address pointers in each descriptor enables the adapter port to be compatible with various types of memory-management schemes.

14.7.2.1. Transmit Descriptor 0

TDES0 contains transmitted frame status and descriptor ownership information.

TDES0										Access = h					Regoffset =					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16					
OWN	Rsv																			
Default value after reset = h																				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	JBT	Rsv		LO	NC	LC	EC	HF	CC				Rsv	UF	DE
Default value after reset = h															

Bit Number	Mnemonic	Description
Bit 31	OWN	Own Bit When set, indicates that the descriptor is owned by the 21143. When cleared, indicates that the descriptor is owned by the host. The 21143 clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the first descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between the 21143 fetching a descriptor and the driver setting an ownership bit.
Bits 30:16	Rsv	Reserved.
Bit 15	ES	Error Summary Indicates the logical OR of the following bits: TDES0<1> Underflow error TDES0<8> Excessive collisions TDES0<9> Late collision TDES0<10> No carrier TDES0<11> Loss of carrier TDES0<14> Transmit jabber timeout summary
Bit 14	TJT	Transmit Jabber Timeout - Reserved



Bits 13:12	Rsv	Reserved
Bit 11	LO	Loss of Carrier When set, indicates loss of carrier during transmission. Not valid in internal loopback mode (CSR6<11:10>=01).
Bit 10	NC	No Carrier When set, indicates that the carrier signal from the transceiver was not present during transmission. Not valid in internal loopback mode (CSR6<11:10>=01).
Bit 9	LC	Late Collision When set, indicates that the frame transmission was aborted due to collision occurring after the collision window of 64 bytes. Not valid if underflow error (TDES0<1>) is set.
Bit 8	EC	Excessive Collisions When set, indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame.
Bit 7	HF	Heartbeat Fail This bit is effective only in 10BASE-T/AUI operating mode. When set, this bit indicates a heartbeat collision check failure (the transceiver failed to return a collision pulse as a check after the transmission). For transceivers that do not support heartbeat collision check, heartbeat fail is set but is not valid. This bit is not valid if underflow error (TDES0<1>) is set. On the second transmission attempt, after the first transmission was aborted due to a collision, the 21143 does not check heartbeat fail (TDES0<7>) and is reset.
Bit3 6:3	CC	Collision Count This 4-bit counter indicates the number of collisions that occurred before the frame was transmitted. Not valid when the excessive collisions bit (TDES0<8>) is also set.
Bit 2	Rsv	Reserved
Bit 1	UF	Underflow Error When set, indicates that the transmitter aborted the message because data arrived late from memory. Underflow error indicates that the 21143 encountered an empty transmit FIFO while transmitting a frame. The transmission process enters the suspended state and sets both transmit underflow (CSR5<5>) and transmit interrupt (CSR5<0>).
Bit 0	DE	Deferred When set, indicates that the 21143 had to defer while ready to transmit a frame because the carrier was asserted.

MAC ETHERNET INTERFACE (LAN)

14.7.2.2. Transmit Descriptor 1

.

TDES1																Access = h		Regoffset =	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
IC	LS	FS	Rsv		AC	TER	TCH	DPD	Rsv	TSB2									
Default value after reset = h																			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSB2					TSB1										
Default value after reset = h															

Bit Number	Mnemonic	Description
Bit 31	IC	Interrupt on Completion When set, the 21143 sets transmit interrupt (CSR5<0>) after the present frame has been transmitted. It is valid only when last segment (TDES1<30>) is set or when it is a setup packet.
Bit 30	LS	Last Segment When set, indicates that the buffer contains the last segment of a frame.
Bit 29	FS	First Segment When set, indicates that the buffer contains the first segment of a frame.
Bits 28 - 27	Rsv	Reserved
Bit 26	AC	Add CRC Disable When set, the 21143 does not append the cyclic redundancy check (CRC) to the end of the transmitted frame. This field is valid only when first segment (TDES1<29>) is set.
Bit 25	TER	Transmit End of Ring When set, indicates that the descriptor pointer has reached its final descriptor. The 21143 returns to the root address of the list. This creates a descriptor ring.
Bit 24	TCH	Second Address Chained When set, indicates that the second address in the descriptor is the next descriptor address, rather than the second buffer address. Transmit end of ring (TDES1<25>) takes precedence over second address chained (TDES1<24>).

Bit 23	DPD	Disabled Padding When set, the 21143 does not automatically add a padding field, to a packet shorter than 64 bytes. When reset, the 21143 automatically adds a padding field and also a CRC field to a packet shorter than 64 bytes. The CRC field is added despite the state of the add CRC disable (TDES1<26>) flag.
Bit 22	Rsv	Reserved
Bits 21:11	TSB2	Buffer 2 Size Indicates the size, in bytes, of the second data buffer. If this field is 0, the 21143 ignores this buffer and fetches the next descriptor. This field is not valid if second address chained (TDES1<24>) is set.
Bits 10:0	TSB1	Buffer 1 Size Indicates the size, in bytes, of the first data buffer. If this field is 0, the 21143 ignores this buffer and uses buffer 2.

14.7.2.3. Transmit Descriptor 2

TDES2

Access = h

Regoffset =

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA1															
Default value after reset = h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA1															
Default value after reset = h															

Bit Number	Mnemonic	Description
Bits 31:0	BA1	Buffer Address 1 Indicates the physical address of buffer 1. There are no limitations on the buffer address alignment.



14.7.2.4. Transmit Descriptor 3

TDES3

Access = h

Regoffset =

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA2															
Default value after reset = h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA2															
Default value after reset = h															

Bit Number	Mnemonic	Description
Bits 31:0	BA2	Buffer Address 2 Indicates the physical address of buffer 2. There are no limitations on the buffer address alignment.



15. SERIAL PORT

15.1. INTRODUCTION

The Serial Port of the STPC is a universal asynchronous receiver/transmitter (UART) which is fully programmable by an 8-bit CPU interface. It supports word lengths from five to eight bits, an optional parity bit and one or two stop bits. If enabled the parity can be odd, even or forced to a defined state. A 16-bit programmable baud rate generator and an 8-bit scratch register are included. Eight modem control lines and a diagnostic loop-back mode are provided. Two 16-Byte FIFOs are included, one for transmit and one for receive. Two DMA handshake lines are provided to indicate when the FIFOs are ready to transfer data to the CPU. An interrupt can be generated from any one of ten sources.

Note that on the STPC Vega, only the Rx and Tx Lines are available.

15.2. FUNCTIONAL DESCRIPTION

15.2.1. TRANSMIT OPERATION

Transmission is initiated by writing the data to be sent to the Transmitter Holding Register. The data will then be transferred to the Transmit Shift Register together with a start bit and parity and stop bits as determined by the Line Control Register. The bits to be transmitted are then clocked out of the transmit shift register by the transmit clock (BAUD#) which comes from the baud rate generator.

If enabled, an interrupt will be generated when the Transmitter Holding Register becomes empty.

15.2.2. RECEIVE OPERATION

Data is clocked into the receiver by the receive clock (RCLK). The receive clock should be 16 times the baud rate of the received data. A filter is used to remove spurious inputs which last for less than two periods of RCLK. When the complete word has been clocked into the receiver the data bits are transferred to the Receiver Buffer Register to be read by the CPU. The receiver also checks for a stop bit and for correct parity, as determined by the Line Control Register.

If enabled, an interrupt will be generated when the data has been transferred to the Receiver Buffer Register. Interrupts can also be generated for incorrect parity or a missing stop bit (frame error).

15.2.3. MODEM CONTROL LINES

The output modem control lines, RTS# and DTR#, can be set or cleared by writing to the Modem Control Register. The current status of the input modem control line, DCD#, RI#, DSR# and CTS# can be read from the Modem Status Register. Bit 2 of this register will be set if the RI# modem status line has changed from low to high since the register was last read.

If enabled, an interrupt will be generated when DSR#, CTS#, RI# or CD# are asserted.

Note that the internal UART is not disabled and there may be conflicts with additional external UARTs when implemented.

15.3. SERIAL INTERFACE SIGNALS

SIN1, SIN2, Input Serial input, data is clocked in using RCLK/16.

SOUT1, SOUT2 Output Serial output, data is clocked out using TCLK/16 (TCLK=BAUD#).

DCD1#, DCD2#, Input Data carrier detect, Active low.

RI1#, RI2#, Input Ring indicator, Active low.



SERIAL PORT

DSR1#, DSR2#, Input Data set ready, Active low.

CTS1#, CTS2#, Input Clear to send, Active low.

RTS1#, RTS2#, Output Request to send, Active low.

DTR1#, DTR2#, Output Data terminal ready, Active low.

BAUD# is an internal output transmit timing clock, derived from CLK divided by the value in the divisor latch DLL & DLM.

15.4. REGISTER DESCRIPTION

15.4.1. ADDRESSING

A0-2 and DLAB (Line Control Register bit 7) define which register appears on DA0 - 7. When CE# and WR# are true,

A0-2 and DLAB define which register is to be written with data.

Table 15-1. Serial Port Register Addresses

Address	DLAB	Register Name	Comment
000	0	RBR Receiver buffer	Read Only
000	0	THR Transmitter Holding	Write Only
001	0	IER Interrupt Enable	
010	X	IIR Interrupt Ident	Read Only
010	X	FCR FIFO Control	Write Only
011	X	LCR Line Control	
100	X	MCR Modem Control	
101	X	LSR Line Status	Read Only
110	X	MSR Modem Status	Read Only
111	X	SCR Scratch	
000	1	DLL Divisor Latch (LS)	
001	1	DLM Divisor Latch (MS)	

Note: X = don't care, either 0 or 1.

The first Serial Port is addressed as COM1 at IO address 3F8h, the second Serial Port is addressed as COM2 at IO address 2F8h.



15.4.2. RECEIVER BUFFER REGISTER

This is an 8-bit read only register. This register is updated from the receive shift register at the end of a receive sequence.

This register is accessed only when the bit 7 (DLAB) of the Line control register = 0.

<i>RBR</i>		Access = 3F8h/2F8h				Regoffset = 000h	
7	6	5	4	3	2	1	0
Default value after reset = undefined or 00h							

Programming notes:

If the FIFOs are disabled this register is undefined after reset.

If the FIFOs are enabled this register will return zero after a reset if the receive FIFO is empty.



15.4.3. TRANSMITTER HOLDING REGISTER

This is a 8-bit write only register. Data is held in this register until transferred to the transmitter shift register.

This register is accessed only when the bit 7 (DLAB) of the Line control register = 0.

THR		Access = 3F8h/2F8h				Regoffset = 000h	
7	6	5	4	3	2	1	0
Default value after reset =							



15.4.4. INTERRUPT ENABLE REGISTER

This register is accessed only when the bit 7 (DLAB) of the Line control register = 0.

IER		Access = 3F8h/2F8h				Regoffset = 001h	
7	6	5	4	3	2	1	0
Rsv				EDSSI	ELSI	ETBEI	ERBFI
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-4	Rsv	Reserved. Read as '0'
Bit 3	EDSSI	Enable Modem status interrupt. When set ("1"), an interrupt is generated if D0, D1, D2 or D3 of the Modem Status Register become set.
Bit 2	ELSI	Enable Rx Status Interrupt. When set ("1"), an interrupt is generated if D1, D2, D3 or D4 of the Line Status Register become set.
Bit 1	ETBEI	Enable Tx Holding Register Empty Interrupt. When set ("1"), an interrupt is generated if THRE=1 or the Transmitting Holding Register is empty.
Bit 0	ERBFI	Enable Receiver Buffer Register. When set ("1"), an interrupt is generated if the Receive Buffer contains data.



15.4.5. INTERRUPT IDENTIFICATION REGISTER

This is a 8-bit read only register.

IIR		Access = 3F8h/2F8h				Regoffset = 002h	
7	6	5	4	3	2	1	0
FIFOE		Rsv		ID2	ID1	ID0	INT#
Default value after reset = 01h							

Bit Number	Mnemonic	Description
Bits 7-6	FIFOE	FIFOE. Returns '1' if FIFOs enabled, otherwise '0'.
Bits 5-4	Rsv	Reserved. Always returns 0.
Bit 3	ID2	Interrupt ID Bit 2. If FIFOs disabled returns 0. See Table 15-2 .
Bit 2	ID1	Interrupt ID Bit 1. See Table 15-2 .
Bit 1	ID0	Interrupt ID Bit 0. See Table 15-2 .
Bit 0	INT#	Not interrupt pending. See Table 15-2 .

Table 15-2. Interrupt Priority

Bit 3	Bit 2	Bit 1	Bit 0	Priority	Comment
0	0	0	1		No interrupt pending
0	1	1	0	1	Receiver Line Status
0	1	0	0	2	Receive Data Available or RX FIFO trigger
1	1	0	0	2	Character Timeout Indication
0	0	1	0	3	Transmitter Holding Register Empty
0	0	0	0	4	Modem Status

Programming notes:

Pending Interrupts are cleared by the following actions:

Priority 1) Reading line status register,

Priority 2) Reading receive buffer register,

Priority 3) Reading this register if priority 3 interrupt OR writing to the transmitter holding register,

Priority 4) Reading the MODEM status register.

When multiple interrupts are pending the interrupt line pulses low after each service.



15.4.6. RECEIVE TIMEOUT INTERRUPT

A RX FIFO character timeout can be identified when ID2 is '1'.

A RX FIFO character timeout occurs if all the following apply:

1. There is at least one character in the FIFO.
2. The most recent character was received longer than four character periods ago (inclusive of all start, parity, and stop bits).
3. The most recent CPU read of the FIFO was longer than four character periods ago.

The character timeout is dependent on the RX clock.

A timeout interrupt is cleared by a CPU read from the RX FIFO.

The timeout timer is restarted on receipt of a new Byte from the input shift register, or on a CPU read from the RX FIFO.

15.4.7. TX FIFO INTERRUPT

The Transmitter Holding Register interrupt occurs when the TX FIFO is empty. It is cleared by writing to the Transmitter Holding Register, or by reading from IIR.

The TX FIFO empty interrupt will be delayed one character period minus the last stop bit period whenever; THRE = 1 and there have not been at least two Bytes in the TX FIFO at the same time since the last time THRE = 1. If the TX interrupt is enabled, setting bit 0 of the FCR will generate an immediate interrupt.

15.4.8. FIFO POLLED OPERATION

If the FIFOs are enabled and at least one of the active bits in IER is disabled, then the Serial Port will operate in the FIFO polled mode. Since the TX and RX paths are controlled separately either one or both can be in the polled mode. The application software should check TX and RX status using the LSR.

15.4.9. FIFO CONTROL REGISTER

This is a 8-bit write only register.

FCR		Access = 3F8h/2F8h				Regoffset = 002h	
7	6	5	4	3	2	1	0
RFTL1	RFTL0	Rsv		DMA1	CLRT	CLRR	FIFOE
Default value after reset =							

Bit Number	Mnemonic	Description
Bit 7	RFTL1	RX FIFO trigger level bit 1. See Table 15-3 .
Bit 6	RFTL0	RX FIFO trigger level bit 1. See Table 15-3 .
Bits 5-4	Rsv	Reserved.
Bit 3	DMA1	Set DMA mode 1. This bit determines the DMA mode which the TXRDY and RXRDY pins support. On reset, or when this bit is cleared, the device operates in DMA mode 0. When this bit is set the device operates in DMA mode 1. This bit has no effect unless the FIFOE bit is set as well. TXRDY - Mode 0: Goes active (low) when TX FIFO, or TX holding register, is empty. Becomes inactive when a Byte is written to the TX channel. TXRDY - Mode 1: Goes active (low) when there is at least one unfilled position in the FIFO, becomes inactive when the FIFO is full. RXRDY - Mode 0: Becomes active (low) when there is at least one character in the RX FIFO or the holding register is full. It becomes inactive when there are no more characters in the FIFO or holding register. RXRDY - Mode 1: Becomes active (low) when the RX FIFO trigger level or timeout occurs, goes inactive when the RX FIFO is empty.
Bit 2	CLRT	Clear TX FIFO. Writing a 1 to this bit clears all Bytes in the TX FIFO and resets its counter logic. The output shift register is not affected. This bit is self-clearing.
Bit 1	CLRR	Clear RX FIFO. Writing a 1 to this bit clears all Bytes in the RX FIFO and resets its counter logic. The input shift register is not affected. This bit is self-clearing.
Bit 0	FIFOE	Enable FIFOs. Writing a 1 to this bit enables both the RX and TX FIFOs. When the FIFOs are either enabled or disabled, both the RX and the TX FIFOs are reset. This bit must be a 1 for any of the other bits in the register to have any effect.

Table 15-3. RX FIFO Trigger Level Bit 1

Bit 7	Bit 6	RX FIFO Trigger Level (Bytes)
0	0	01
0	1	04
1	0	08
1	1	14



15.4.10. LINE CONTROL REGISTER

LCR		Access = 3F8h/2F8h				Regoffset = 003h	
7	6	5	4	3	2	1	0
DLAB	SB	SP	EPS	PEN	STB	WLS1	WLS0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	DLAB	Divisor Latch Access Bit. When clear '0', Receive and Transmitter Registers are read/written address 0 and IER register at address 1. When set '1', Divisor Latch LS is read/written at address 0 and Divisor Latch MS read/written at address 1.
Bit 6	SB	Set Break. When set '1', SOUT signal is forced into the '0' state.
Bit 5	SP	Stick Parity. When set '1', Parity bit is forced into a defined state, dependent upon state of EPS, PEN: If EPS = '1' & PEN = '1' Parity bit is set and checked = '0'. If EPS = '0' & PEN = '1' Parity bit is set and checked = '1'.
Bit 4	EPS	Even Parity Select. When set '1' and PEN = "1" an even number of ones is sent and checked. When clear "0" and PEN = "1" an odd number of ones is sent and checked.
Bit 3	PEN	Parity Enabled. When set "1" parity is transmitted and checked. Parity bit is added after the data field and before the STOP bits. When clear "0" parity is neither transmitted or checked.
Bit 2	STB	Number of Stop bits. When set "1" two STOP bits are added after each character is sent, except if character length is 5 then 1½ STOP bits are added. When clear "0" one STOP bit is always added. Only the transmit STOP bits are programmable, the receive stage only expects one STOP bit irrespective of the state of STB.
Bit 1	WLS1	Word Length Select. Transmitted and Received character size is defined in Table 15-4 .
Bit 0	WLS0	Word Length Select. Transmitted and Received character size is defined in Table 15-4 .

Table 15-4. Word Length Select

Bit 1	Bit 0	Character Size
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits



15.4.11. MODEM CONTROL REGISTER

MCR			Access = 3F8h/2F8h			Regoffset = 004h	
7	6	5	4	3	2	1	0
Rsv			Loop	Rsv		RTS	DTR
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-5	Rsv	Reserved. Read as '0'
Bit 4	Loop	Loop back mode. When set '1' the following conditions are implemented: 1) SOUT is forced to '1'. 2) SIN is disconnected from the Receive input shift register. 3) Receive shift register input is connected to Transmitter shift register output. 4) The Modem status signals are disconnected (CTS#, DSR#, DCD#, RI#). 5) The Modem control signals are connected to modem status inputs (RTS to CTS and DTR to DSR). When clear '0', Modem and control/status signals SIN/SOUT are as normal.
Bits 3-2	Rsv	Reserved.
Bit 1	RTS	Control Signal. This signal controls the state of the RTS# output even in loop mode. When RTS = '0' RTS# = '1'. When RTS = '1' RTS# = '0'.
Bit 0	DTR	Control Signal. This signal controls the state of the DTR# output even in loop mode. When DTR = '0' DTR# = '1'. When DTR = '1' DTR# = '0'.



15.4.12. LINE STATUS REGISTER

This is a 8-bit read only register.

LSR

Access = 3F8h/2F8h

Regoffset = 005h

7	6	5	4	3	2	1	0
FIFOERR	TEMT	THRE	BI	FE	PE	OE	DR
Default value after reset = 60h							

Bit Number	Mnemonic	Description
Bit 7	FIFOERR	RX Data Error in FIFO. This bit is set to '1' when there is at least one PE, FE, or BI in the RX FIFO. It is cleared by a read from the LSR register, if there are no subsequent errors in the FIFO.
Bit 6	TEMT	Transmitter Empty. If the FIFOs are disabled, this bit is set to '1' whenever the transmitter holding register and the transmitter shift register are empty. If the FIFOs are enabled, this bit is set whenever the TX FIFO and the transmitter shift register are empty. In both cases this bit is cleared when a Byte is written to the TX data channel.
Bit 5	THRE	Transmitter Holding Register Empty. If the FIFOs are disabled, this bit is set to '1' whenever the transmitter holding register is empty and ready to accept new data, this bit is cleared when the data is transferred to the transmitter shift register. If the FIFOs are enabled, this bit is set to '1' whenever the TX FIFO is empty. It is cleared when at least one Byte is written to the TX FIFO.
Bit 4	BI	Break Interrupt. If the FIFOs are disabled, this bit is set whenever the SIN is held in the 0 state for more than a transmission time (START bit + DATA bits + PARITY + STOP bits). BI is reset by the CPU reading this register. If the FIFOs are enabled, this error is associated with the corresponding character in the FIFO. The error is flagged when this Byte is at the top of the FIFO. When break occurs only one zero character is loaded into the FIFO, the next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.
Bit 3	FE	Framing Error. If the FIFOs are disabled, this bit is set if the received data did not have a valid STOP bit, FE is reset by the CPU reading this register. If the FIFOs are enabled, the state of this bit is revealed when the Byte it refers to is at the top of the FIFO.
Bit 2	PE	Parity Error. If the FIFOs are disabled, this bit is set if the received data does not have a valid parity bit, PE is reset by the CPU reading this register. If the FIFOS are enabled, the state of this bit is revealed when the Byte it refers to is at the top of the FIFO.

Bit 1	OE	<p>Overrun Error. If the FIFOs are disabled, this bit is set if the receive buffer was not read by the CPU before new data from the receive shift register over wrote previous contents. OE is cleared when the CPU reads this register.</p> <p>If the FIFOs are enabled, an overrun error occurs when the RX FIFO is full and the RX shift register becomes full. OE is set as soon as this happens. The character in the shift register is then overwritten, but is not transferred to the FIFO.</p>
Bit 0	DR	<p>Data Ready. This bit is set whenever the receive buffer is full., or by a Byte being transferred into the FIFO. DR is cleared by the CPU reading the receive buffer, or by reading all of the FIFO Bytes.</p> <p>This bit is also cleared whenever the FIFO enable bit is changed.</p>



15.4.13. MODEM STATUS REGISTER

This is a 8-bit read only register.

MSR		Access = 3F8h/2F8h				Regoffset = 006h	
7	6	5	4	3	2	1	0
DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
Default value after reset =							

Bit Number	Mnemonic	Description
Bit 7	DCD	Data Carry Detect. When Loop = '0' this is the complement of input signal DCD#. When Loop = '1' this is equal to OUT2.
Bit 6	RI	Ring Indicator. When Loop = '0' this is the complement of input signal RI#. When Loop = '1' this is equal to OUT1.
Bit 5	DSR	Data Set Ready. When Loop = '0' this is the complement of input signal DSR#. When Loop = '1' this is equal to DTR.
Bit 4	CTS	Clear To Send. When Loop = '0' this is the complement of input signal CTS#. When Loop = '1' this is equal to RTS.
Bit 3	DDCD	Delta Data Carry Detect. This bit is set ('1') if the state of DSR has changed since this register was last read.
Bit 2	TERI	Trailing Edge Ring Indicator. This bit is set if the RI# input changes from '0' to '1' since this register was last read.
Bit 1	DDSR	Delta Data Set Ready. This bit is set ('1') if the state of DSR has changed since this register was last read.
Bit 0	DCTS	Delta Clear to Send. This bit is set ('1') if the state of CTS has changed since this register was last read.

Programming notes:

After reset, Bits 7-4 are inputs, Bits 3-0 = '0' and can be written to.

A modem status interrupt can be cleared by writing '0' or set by writing '1' to this register.Scratch Register



15.4.14. SCRATCH REGISTER

This is a general purpose read/write register.

SCR		Access = 3F8h/2F8h				Regoffset = 007h	
7	6	5	4	3	2	1	0
Default value after reset = undefined							



15.4.15. DIVISOR LATCH (LS) - DIVISOR LATCH (MS)

These registers are accessed only when bit 7 (DLAB) of the Line control register = 1.

DLL		Access = 3F8h/2F8h				Regoffset = 000h	
7	6	5	4	3	2	1	0
Default value after reset = undefined							

DLM		Access = 3F8h/2F8h				Regoffset = 001h	
7	6	5	4	3	2	1	0
Default value after reset = undefined							

Programming notes:

Table 15-5. lists the possible DLL & DLM settings and their respective Baud rates.

Table 15-5. Decimal Divisor

BAUD	DLM Setting	DLL Setting
110	23	82
300	0D	05
1200	03	41
2400	01	A1
4800	00	D0
9600	00	68
19200	00	34
38400	00	1A
57600	00	11



15.5. SPECIAL FEATURES

15.5.1. TRANSMIT MACHINE TIMING

The TXM (Transmit Machine) starts after 2-3 baud clocks from the time the Transmitter Holding Register is written. The SOUT goes low 7-8 baud clocks from the Transmitter Holding Register being written.

15.5.2. THR EMPTY INTERRUPT TIMING

A Transmitter Holding Register Empty interrupt will be generated 17-18 clocks after data has been written to the Transmitter Holding Register, providing that the Transmit Machine was idle when the data was written.

If the Transmitter Holding Register is empty when the Transmitter Holding Register Empty interrupt is enabled an interrupt will be generated immediately.

15.5.3. FIFO RESET TIMING

When using bits 0-3 of the FIFO Control Register to reset the FIFOs the following timing restrictions apply:

FCR0 - Both FIFOs are reset by the master reset (MR), and are held reset unless FCR0 is set to 1.

FCR1 - The RXFIFO clear requires at least one RCLK period to complete the reset and clear itself.

FCR2 - When set to 1, the TXFIFO clear holds the transmit FIFO reset until the leading edge of the next write strobe to the transmit FIFO, or the next read strobe to the IIR.

16. I²C BUS CONTROLLER

16.1. INTRODUCTION

The I²C (Inter-Integrated Circuit) Controller built into the STPC device provides a two-wire communication link between the STPC and external integrated circuits. The I²C Bus physically consists of two active bi-directional wires, SDA (Serial DATA line) and SCL (Serial CLock line), together with a ground connection. Every device connected to the I²C Bus is assigned a unique address and each can act as a receiver and/or a transmitter, depending on the functionality.

Both the SDA and SCL lines are connected to a positive supply voltage via a pull-up resistor. When the bus is free, both lines are HIGH. The output stage of the I²C Controller interface connected to the bus must have an open-drain or open-collector to perform the wired-AND function.

The I²C bus allows the parallel microprocessor to interface to other I²C compatible peripherals. Data is transmitted to and received from the I²C bus via a buffered interface. Control and status information is relayed through a set of registers via the STPC RBI Interface.

The I²C protocol specifies that the device currently issuing commands on the I²C bus is the Bus Master and that all others at this time are Bus Slaves. This Multi-Master Bus arrangement means therefore that more than one device capable of initiating data transfers can be connected to it.

The I²C bus thus requires a minimum of hardware to provide an economical system which is widely accepted as an industry standard.

16.2. FEATURES

- Both Master and Slave functions are supported.
- Multi-Master operation supported.
- Programmable depth transmitter and receiver FIFO registers.
- Master can address both a 7-bit and a 10-bit Slave.
- Master performs arbitration and clock synchronization.
- Wide range of clock frequencies supported, including Standard (100 kHz) and Fast (400 kHz). High Speed modes also supported (2.75 Mbit/s maximum).
- Software Reset supported
- Interrupt and polling operations both supported.
- Spike rejection of 90 ns in Slow/Fast mode.
- Slave can be addressed via both standard 7-bit and extended 10-bit addressing schemes.

16.3. FUNCTIONAL DESCRIPTION

To accomplish a data transfer, the I²C Controller can be operated in one of four modes, as listed below.

Table 16-1. Modes of Operation

Mode	Definition
Master - Transmit	<ul style="list-style-type: none">• I²C Bus Controller acts as a master• Used for a write operation• I²C Bus Controller sends the data• I²C Bus Controller is responsible for clocking• Slave Device is in slave-receive mode
Master - Receive	<ul style="list-style-type: none">• Bus Controller acts as a master• Used for a read operation• Bus Controller receives the data• Slave Device is in slave-transmit mode
Slave - Transmit	<ul style="list-style-type: none">• Bus Controller acts as a master• Used for a read operation• Bus Controller sends the data• Master is responsible for clocking• Slave Device is in slave-transmit mode
Slave - Receive	<ul style="list-style-type: none">• Bus Controller acts as a master• Used for a read operation• Bus Controller receives the data• Master is responsible for clocking• Slave Device is in slave-transmit mode

When the I²C Controller is in IDLE mode (neither receiving or transmitting serial data), the controller defaults to Slave Receive mode. This allows the controller to monitor the bus and receive the slave address that might be intended for it. If the address received by the I²C controller (can be either the 7 bit or the 10 bit) matches the programmed slave address in the COMMAND register, the controller either remains in the slave Receive mode or transitions to the Slave-Transmit mode. This is determined by the Read/Write (R/W#) bit (the least significant bit of the byte containing the Slave Address). If the R/W# bit is low, the master initiating the transaction intends to do a write and the I²C Bus Controller remains in the Slave-Receive mode. If the R/W# bit is high, the initiating master wants to do a read and the transitions to the Slave-Transmit mode.

When the I²C Controller wants to initiate a read or a write on the I²C Bus, the I²C Bus Controller should be programmed to the Master Transmit mode. To write data, the controller remains in the master-transmit mode after the Address Transfer has completed. To read data, the controller transmits the start address and transfers to the Master Receive mode.

16.4. I2C BUS OPERATION

The I²C Bus Controller transfers data in one byte increments and always follows the sequence:

1. START
2. 7-bit Slave Address
3. R/W# Bit
4. Acknowledge Pulse
5. Eight Bits of Data



6. Ack/Nack Pulse
7. Repeat of Steps 5 and 6 for Required Number of Bytes
8. Repeated START (Repeat Step 1) or STOP

16.5. SERIAL CLOCK LINE (SCL) GENERATION

The I²C Controller is required to generate the I²C clock output when in master mode (either receive or transmit). SCL clock generation is accomplished through the use of the CLOCKCONTROL register value, which is programmed at initialization. The CLOCKCONTROL value is used in the following equation to determine the SCL transition period.

$$\text{SCL Period} = (\text{SCL Low} + \text{SCL High}) \text{ Decimal Value} * \text{PCI Bus Clock Period}$$

When wait states are inserted or synchronization with another master is necessary, the I²C controller performs the necessary clock synchronization. The CLOCK CONTROL register provides a simple method for determining I²C clock frequencies. The table below details sample programming.

Table 16-2. Clock Control Programming

PCI Bus Frequency	I ² C Clock Frequency = [1/(SCL Low Period + SCL High Period)] * PCI Bus Frequency	CLOCK CONTROL Reg Value			
		Hexadecimal Value	Setup Time	SCL Low	SCL High
33 MHz	106 kHz	9D009D9D ₁₆	157	157	157
	412 kHz	14002828 ₁₆	20	40	40
25 MHz	80 kHz	9D009D9D ₁₆	157	157	157
	312 kHz	14002828 ₁₆	20	40	40

16.6. I²C CONTROLLER MASTER OPERATION

To transfer data on the I²C bus through the I²C Bus Controller, enable either the Master Fast Mode (400 kHz) or the Standard Mode (100 kHz). If Interrupt generation is required, enable the interrupt enable bit in the COMMAND register. After the start Command, the I²C Controller will wait until the I²C Bus is busy. After the Bus becomes free, which is detected when ISCL is high and SDA goes from low to high, the I²C Controller will start transferring data serially on the SDA line.

If, after a start or restart, the least significant bit in the first byte (the address) is 0, this indicates a "WRITE", and the I²C Controller will act as a Master transmitter for the whole transaction and will pull the SDA line high during the acknowledge SCL pulse to allow Slave to assert acknowledgement. If the least significant bit in the first byte after the start or restart is 1, this indicates a "READ". After the transmission of the first byte, the I²C Controller then acts as a Master receiver and stores the data in the Receiver FIFO. In this case, the I²C controller will give acknowledgement after successfully receiving one byte.

If the Master loses arbitration, it returns to the Slave receiver Mode.

16.7. I²C CONTROLLER SLAVE OPERATION

For the I²C Controller to act as a Slave on the I²C Bus, it is necessary to program the proper slave address



in the configuration register. For a 10-bit address, also program upper bits in the Slave Upper Address Bit register. For any cycle initialized on the I²C Bus, the I²C Controller monitors the start or restart condition on the I²C Bus, after which it obtains the address information by collecting the serial bits on the I²C bus. If the address is the same as that programmed in the configuration register, the I²C Controller issues an acknowledgement (during the acknowledgement clock pulse) by asserting the SDA line low.

After issuing the acknowledgement, the I²C Controller acts as a Slave transmitter for a read command and acts as a Slave receiver for a write command. If in a Master mode the frequency of SCL is high, or in a Slave mode there is a lag in reading or storing the data, then the I²C Controller can insert a wait state by holding down the SCL line. The I²C Controller goes into the idle state after the stop is detected on the I²C Bus.

16.8. START AND STOP BUS STATES

The I²C bus uses START and STOP transaction bus states at the beginning and end of the transfer of one byte or an unlimited number of bytes on the bus. The I²C Controller uses the **mstStart** and **mstStop** bits in the COMMAND register to:

- Initiate an additional byte transfer
- Initiate a START condition on the I²C Bus
- Enable Data Chaining (Repeated START)
- Initiate a STOP condition on the I²C Bus

The **mstStart** and **mstStop** COMMAND register bit definitions are given in the following table.

Table 16-3. START and STOP Bit Definitions

mstStart	mstStop	Function
0	0	Reserved
0	1	This configuration asserts STOP. In Master Transmitter or Receiver mode, the I ² C Controller asserts a STOP on the I ² C Bus, after completing data transfer.
1	0	<ul style="list-style-type: none">• In the Master Transmitter Mode, the Master will start transmitting. If the transmit FIFO becomes empty, the Controller waits for either a start command or a stop command or until the transmitter FIFO contains at least one byte.• In the Master Receiver Mode, the Master will start receiving data. If the transmitter FIFO is empty or the receiver FIFO is full, the I²C Controller will wait either for the start command or the stop command or until the transmitter FIFO contains at least one byte or the receiver FIFO has one empty location to fill.
1	1	For this configuration the Master will start and will automatically generate a STOP when either the transmitting FIFO is empty in the transmitting mode or the receiving FIFO is full the receiving mode.

16.9. GLITCH SUPPRESSION LOGIC

The I²C Bus Controller has built-in glitch suppression logic. Glitches are suppressed according to the SCL



clock speed mode (Fast/High Speed) and the PCI clock period. For example, with a 33 MHz (30 ns period) PCI clock, glitches of 90 ns or less are suppressed in Standard/Fast mode of operation of the I²C bus. In the High speed mode of operation, glitches of 30 ns or less can be suppressed.

16.10. RESET CONDITIONS

The I²C controller is RESET with sysReset, or by setting the soft Reset bit in the control register. Software is responsible for ensuring that the I²C Controller is not busy before asserting RESET. Software is also responsible for ensuring that the I²C bus is idle when the Controller is enabled after RESET. When directed to RESET, the I²C Controller returns to its default RESET condition, when all registers including the Slave Address are RESET. When the soft Reset bit in the CONTROL Register is set, only the I²C Controller is reset. When resetting the I²C Controller with the CONTROL register controller reset, use the following guidelines:

In the CONTROL register, set the softReset bit, the TRANSMIT FIFO flush bit, the RECEIVE FIFO flush bit and clear the remainder of the register.

16.11. MASTER OPERATION

When software initiates a read or write operation on the I²C bus, the I²C Controller switches from the default Slave-Receive mode to the Master-Transmit mode. The start pulse is sent followed by the 7-bit slave address and the R/W# bit. After the Master receives an acknowledge, the I²C Controller has the option of two Master modes:

- Master-Transmit - The I²C Controller writes data
- Master-Receive - The I²C Controller reads data

The I²C Controller processor initiates a master transaction by writing to the COMMAND register. Data is read and written from the I²C Controller through the I/O-mapped setup registers.

Table 16-4. Master Transactions

I ² C Master Action	Mode of Operation	Definition
Generate clock output	Master-Transmit Master-Receive	<ul style="list-style-type: none"> • The Master always drives the SCL line. • The CLOCKCONTROL register is written. • Set the mstStart bit in the COMMAND register. • The SCL Clock is driven soon after Master starts the operation
Write target (slave address + R/W# Bit) TRANSMIT BUFFER	Master-Transmit Master-Receive	<ul style="list-style-type: none"> • Write slave address in TRANSMIT BUFFER before START condition is enabled. • 8 bits from TRANSMIT BUFFER are sent on SDA bus after START. • SCL Clock is driven soon after Master starts operation • If R/W# Bit is low, the Master remains a Master-Transmitter. If high, the Master changes to a Master Receiver. • See START AND STOP BUS STATES, Section 16.8.

Table 16-4. Master Transactions

I²C Master Action	Mode of Operation	Definition
Signal START Condition	Master-Transmit Master-Receive	<ul style="list-style-type: none"> • See “Generate clock output” above. • Performed after the target slave address and the R/W# bit are in the TRANSMIT BUFFER. • Set the mstStart bit in the COMMAND Register to generate START.
Initiate first data byte transfer	Master-Transmit Master-Receive	<ul style="list-style-type: none"> • Write DATA byte to TRANSMIT BUFFER. • I²C Bus Controller waits for byte to be written to the TRANSMIT FIFO. After it is written, it transmits the byte & sets the Byte Transmitted bit in the STATUS Register.
Arbitrate for I ² C Bus	Master-Transmit Master-Receive	<ul style="list-style-type: none"> • If two or more Masters signal a start within the same clock period, arbitration must occur. • The I²C Bus Controller arbitrates for as long as necessary. Arbitration takes place during slave address, R/W# bit, and data transmission and continues until all but one Master loses the bus. No data is lost during arbitration. • If the I²C Bus Controller loses arbitration, it sets the Master Arbitration Loss Detect bit in the STATUS Register and changes to the Slave-Receive (default) mode.
Write one data byte to the TRANSMIT BUFFER	Master-Transmit Master-Receive	<ul style="list-style-type: none"> • Data transmit mode of I²C Master operation. • Occurs when the data byte to be transferred is written to the TRANSMIT FIFO. If enabled, the Transmit FIFO Empty, or Byte transmitted, is signalled to the Interrupt Generation Unit. • Write one data byte to the TRANSMIT FIFO and set the appropriate START/STOP bit combination. Eight bits are written on the serial bus followed by a STOP if requested (number of data bytes to be transmitted is programmed to be one in the BYTECOUNT register).
Wait for Acknowledge from Slave-Receiver	Master-Transmit Master-Receive	<ul style="list-style-type: none"> • As a Master-Transmitter, the I²C Bus Controller generates the clock for the acknowledge pulse. The I²C Bus Controller is responsible for releasing the SDA line to allow Slave-Receiver Acknowledge transmission.

Table 16-4. Master Transactions

I ² C Master Action	Mode of Operation	Definition
Read one byte of I ² C Data from the RECEIVE FIFO	Master-Transmit Master-Receive	<ul style="list-style-type: none"> Data receive mode of I²C Master operation. Eight bits are read from the serial bus, collected in the shift register then transferred to the RECEIVE FIFO after the Ack/Nack bit is read. Byte Received bit in the STATUS register is set. Reads from the RECEIVE FIFO can be done when the RECEIVE FIFO is not empty, or Byte Received bit in the STATUS register is set. If enabled, a RECEIVE FIFO Full, or byte Received, is signalled to the Interrupt Generation unit. When the RECEIVE FIFO is read, if the mstTmtNoAck Status is clear (indicating Ack), the Master-writes the acknowledge control bit (forceNotAck in the Command Register). The next byte is read if the number of bytes to be read specified in the BYTECOUNT Reg is not complete. If the mstTmtNoAck Status bit is set (indicating Nack), the STOP bit in the COMMAND register is set, then the last data byte has been read into the RECEIVE FIFO and the I²C Bus Controller sends the STOP. If the mstTmtNoAck Status bit is set (indicating Nack), the number of bytes to be read specified in the BYTECOUNT Register is not complete, but the STOP bit is clear, then the microprocessor has two options: <ol style="list-style-type: none"> Set the START bit, and write a new target address to the TRANSMIT FIFO, which sends a repeated start condition. Set the STOP bit which sends a STOP only. If enabled, a Master transmission "NoAck" is signalled to the Interrupt Generation unit.
Transmit Acknowledge to Slave-Transmitter	Master-Transmit Master-Receive	<ul style="list-style-type: none"> As a Master-Receiver, the I²C Bus Controller generates the clock for the acknowledge pulse. The I²C Bus Controller is also responsible for driving the SDA line during the Ack cycle. <p>If the next data byte is to be the last transaction, this sets the acknowledge control bit (forceNotAck in the Command Register) for Nack 1 generation.</p>
Generate a Repeated START to chain I ² C transactions	Master-Transmit Master-Receive	<ul style="list-style-type: none"> If data chaining is desired, a repeated START condition is used instead of a STOP condition. This occurs after the last data byte of a transaction has been written to the bus. The CPU writes the next target slave address and the R/W# bit to the TRANSMIT FIFO, set the START bit.

Table 16-4. Master Transactions

I ² C Master Action	Mode of Operation	Definition
Generate a STOP	Master-Transmit Master-Receive	<ul style="list-style-type: none"> Generated after the last data byte on the bus has been transferred. Generates a STOP condition by setting the STOP bit in the COMMAND Register.

Note: Nack implies DO NOT ACKNOWLEDGE

16.12. SLAVE OPERATION

Details of Slave operation are given in the following table.

Table 16-5. Slave Transactions

I ² C Slave Action	Mode of Operation	Definition
Slave-Receive (default mode)	Slave-Receive only	<ul style="list-style-type: none"> The I²C Bus Controller monitors all slave address transactions. The I²C Bus Controller Slave Address, Slave Address Set bit, Slave Address Size(7/10 bit) in the Command Register should be programmed. The I²C Bus Controller monitors the bus for START conditions. When a START is detected, the interface reads the first eight bits and compares the most significant seven bits with the 7-bit I²C Slave Address in the Command Register. If there is a match, the I²C Bus Controller sends an Acknowledge (Ack) signal. If the eighth bit of the first byte (R/W# bit) is low, the I²C Bus Controller stays in Slave-Receive mode. If the R/W# bit is high, the I²C Bus Controller changes to the Slave-Transmit mode. If the slave Address is 10-bit, then if the most significant seven bits matches 11110XX, where xx denotes the two most significant bits of the 10-bit address, then a Slave Address Hit is generated. If the R/W bit is 'zero', then the second byte contains the remaining eight bits (XXXXXXXX) of the 10-bit address. If the R/W bit is 'one', then the next byte contains data transmitted from a Slave to a Master.
Setting the Slave Address Detected bit	Slave-Receive Slave-Transmit	<ul style="list-style-type: none"> Indicates the interface has detected an I²C operation that addresses the I²C Controller. The I²C Controller does not respond to a General Call Address.

Table 16-5. Slave Transactions

I ² C Slave Action	Mode of Operation	Definition
Read one byte of I ² C Data from the RECEIVE FIFO	Slave-Receive only	<ul style="list-style-type: none"> Data receive mode of I²C slave operation. Eight bits are read from the serial bus into the shift register. When a full byte has been received and the slave Ack/Nack bit has completed, the byte is transferred from the shift register to the RECEIVE FIFO, and byte received bit is set in the STATUS Register. When the RECEIVE FIFO is Full, STATUS register bit (receive FIFO FULL) is set. If enabled, the RECEIVE FIFO Full is signalled to the Interrupt generator unit. This condition causes the I²C Bus Controller to insert wait states. When the RECEIVE FIFO is read by the CPU. This causes the I²C Bus Controller to stop inserting wait states and let the Master Transmitter write the next piece of information.
Transmit Acknowledge to Master-Transmitter	Slave-Receive only	<ul style="list-style-type: none"> As a Slave-Receiver, the I²C Bus Controller is responsible for pulling the SDA line low to generate the Ack pulse during the high SCL period. The Slave Ack/Nack Control bit controls the Ack data the I²C Bus Controller drives on the SDA line.
Write one byte of I ² C data to the TRANSMIT FIFO	Slave-Transmit only	<ul style="list-style-type: none"> Data transmit mode of I²C slave operation. Occurs when the TRANSMIT FIFO Empty bit is set. If enabled, the TRANSMIT FIFO Empty is signalled to the Interrupt Generator. The CPU writes a data byte to the TRANSMIT FIFO to start transmission.
Wait for Acknowledge from Master-Receiver	Slave-Transmit only	<ul style="list-style-type: none"> As a slave-transmitter, the I²C Bus Controller is responsible for releasing the SDA line to allow the Master-Receiver to pull the line low for the Ack signal.

16.13. SLAVE MODE PROGRAMMING EXAMPLES

16.13.1. Initialize Controller

1. Write CLOCKCONTROL: Set clock count
2. Write COMMAND: Set Slave Address, Slave Address Set, Slave Acknowledge
3. Write CONTROL: Enable all interrupts, program FIFO Depth.

16.13.2. Write One byte as a Slave

1. Wait for Slave Address Detected interrupt. Read STATUS Register: Slave Address Hit (set), Slave Transmit/Receive_ (1), Ack/Nack (Clear - Ack)
2. Write to TRANSMIT FIFO: Load data byte to transmit FIFO

3. Wait for TRANSMIT FIFO Empty (interrupt/poll Status Register). Read STATUS: TRANSMIT FIFO Empty (set), slave Ack/Nack (set - indicates last byte write), Slave Transmit/Receive bit (1). Reading Status Register clears Interrupt.
4. Read STATUS: Slave STOP Detected (set)
5. Wait for STOP on I²C bus.

16.13.3. Read Two Bytes as a Slave

1. Wait for Slave Address Hit interrupt. Read STATUS: Slave Address Hit (set), R/W# bit (0)
2. Wait for interrupt/poll STATUS Register due to Byte Received. Read STATUS: Byte Received (set), Ack/Nack (clear), Slave Transmit/Receive_ bit (0) Interrupt is Cleared by reading STATUS Register. Read RECEIVE FIFO to get the data.
3. Read byte two on the I²C bus.
4. Wait for interrupt/poll STATUS Reg. due to Byte Received. Read STATUS: Byte Received (set), Ack/Nack (clear), R/W# bit (0). Interrupt is Cleared by reading STATUS Register. Read TRANSMIT FIFO to get the data.
5. Wait for STOP on I²C bus.

16.14. MASTER PROGRAMMING EXAMPLES

16.14.1. Initialize Controller

1. Write CLOCKCONTROL: Set clock count
2. Write COMMAND: Set Master START/STOP
3. Write CONTROL: Enable all necessary interrupts, FIFO depths (default 16)

16.14.2. Write One Byte as a Master

1. Write to TRANSMIT FIFO: Target Slave Address and R/W# bit (0 for write), one data byte to be transmitted.
2. Write COMMAND: Set START bit, set STOP bit.
3. Write BYTECOUNT: Device Address+Bytes to be transmitted.
4. Wait for TRANSMIT FIFO Empty interrupt/poll in the STATUS Register. When interrupt arrives: Read status register: TRANSMIT FIFO Empty (set), Interrupt gets cleared on reading the STATUS Register. Note that the Arbitration Loss Detected bit may be set.
5. Since STOP bit was set, the Master will send STOP on I²C bus after transmit of all bits is complete.
6. Wait for I²C Bus to get free by polling the STATUS Register bit 0.

16.14.3. Read One Byte as a Master

1. Write TRANSMIT FIFO: Target Slave Address and R/W# bit (1 for read)
2. Write COMMAND: Set START bit, STOP bit to Initiate I²C bus transfer
3. Write BYTECOUNT: Number of Data Bytes to be received.
4. Wait for TRANSMIT FIFO Transmit Empty interrupt. When interrupt arrives: Read status register: TRANSMIT FIFO Empty (set), R/W# bit (set). Interrupt gets cleared on reading the STATUS Register.
5. Wait for Byte Received interrupt. If the Receive FIFO size has been programmed to one byte, then wait for RECEIVE FIFO full interrupt. Controller sends STOP on the I²C bus after the number of bytes specified in the BYTECOUNT register have been read. Read status register: TRANSMIT FIFO Receive Full (set), Byte Received (set). Interrupt gets cleared on reading the STATUS Register. Read RECEIVE FIFO data.

16.15. CONFIGURATION REGISTERS

The I²C Controller contains the following I²C Control Registers, configured through the the STPC RBI bus using the following pseudo code:

```
I2C_SLCT_LSB    0x07
```

```
I2C_SLCT_MSB    0x00
```

```
IOWrite8(0x22,0x10);
```

```
IOWrite8(0x23,I2C_SLCT_LSB);
```

```
IOWrite8(0x22,0x11);
```

```
IOWrite8(0x23,I2C_SLCT_MSB);
```

16.16. I²C CONTROLLER CONTROL REGISTERS

This section describes the I²C Controller Control Registers, as listed in the following table.

Table 16-6. I²C Controller Control Registers

Register Name	Offset	Width
Buffer Receive	0x0	8-bit
Buffer Transmit	0x0	8-bit
Status	0x4	32-bit
Command	0x8	32-bit
Control	0xC	32-bit
Clock Control	0x10	32-bit
Scratch	0x14	32-bit
Byte Count	0x18	32-bit

16.16.1. Buffer Receive Register

The data stored in the read only Receiver Buffer is delivered to the host when it initiates a read cycle.

Buffer_Receive

Index 00h

7	6	5	4	3	2	1	0
Received byte from receive FIFO							
Default value after reset = 00h							

16.16.2. Buffer Transmit Register

The byte to be transmitted on the I²C bus is written into this register. Normally, the CPU writes to this register when the I²C Controller is acting as either a Master Transmitter or a Slave Transmitter. The transmit buffer FIFO empty interrupt is cleared by writing into the transmit buffer FIFO.

Buffer_Transmit							Index 00h
7	6	5	4	3	2	1	0
Byte of data to transmit							
Default value after reset = 00h							



16.16.3. Status Register

Status

Offset 0x04h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv							MD	POI	SAH	STNA	MLA	MTNA	MANA	BR	BT
Power-ON default: 0x0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RC				TC				Rsv	RE	RFF	TFE	TFF	MSO	SFTR	BB
Power-ON default: 004a0h															

Bit Number	Mnemonic	Description
Bits 31:25	Rsv	Reserved:
Bit 24	MD	MstDone: This read only bit is set when the Master has completed transmission/reception.
Bit 23	POI	PollOrInt: If this read/clear bit is set, at least one condition for the generation of an interrupt has been met.
Bit 22	SAH	SlvAddrHit: This read/clear bit is set whenever the Slave is addressed and the command is write.
Bit 21	STNA	SlvTmtNoAck: This read/clear bit is set when acting as a Slave transmitter and no acknowledgement is received.
Bit 20	MLA	MstLostArbitration: This read/clear bit is set when the I ² C Controller is acting as a Master and arbitration has been lost.
Bit 19	MTNA	MstTmtNoAck: This read/clear bit is set when the I ² C Controller is acting as a Master and no acknowledgement has been received.
Bit 18	MANA	MstAddrNoAck: If this read/clear bit is set, it indicates that the Master, after transmitting an address, does not receive an acknowledgement.
Bit 17	BR	ByteRecieved: This read only bit is set for each byte received in the receive FIFO.

I²C BUS CONTROLLER

Bit Number	Mnemonic	Description
Bit 16	BT	ByteTransmitted: This read only bit is set for each byte transmitted on the I ² C bus from the transmitting FIFO.
Bits 15:12	RC	ReceiveCount: These read only bits indicate the number of bytes in the Receive FIFO.
Bits 11:8	TC	TransmitCount: These read only bits indicate the number of bytes in the Transmit FIFO.
Bit 7	Rsv	Reserved
Bit 6	RE	RcvEmpty: (Receive FIFO Empty) This read only bit is set when the receiver FIFO is found empty when either the Master or Slave is receiving the data.
Bit 5	RF	RcvFull: (Receive FIFO Full) This read only bit is set when the receiver FIFO is found full when either the Master or Slave is receiving the data.
Bit 4	TE	TmtEmpty: (Transmit FIFO Empty) If this read only bit is set it means the transmitter FIFO is found empty when either the Master or Slave is transmitting the data.
Bit 3	TF	TmtFull: (Transmit FIFO Full) If this read only bit is set it means the transmitter FIFO is found full when either the Master or Slave is transmitting the data.
Bit 2	MOS	MasterOrSlave: Master or Slave operation 0: I ² C Controller is acting as a Slave. 1: I ² C Controller is acting as a Master.
Bit 1	STR	SlvTmtRcv: (Slave functions as a Transmitter/Receiver) If this read only bit is 0 it indicates that a Slave is acting as a receiver. If set to a 1 it indicates that a Slave is acting as a transmitter.
Bit 0	I ² CBB	I²CBusBusy: This read only bit is set when a transaction is taking place on the I ² C bus.

16.16.4. Command Register

Command

Offset 0x08h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv					MRT	FNA	SA	SAU			SAL				
Power-ON default: 0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAL		SAS	SAM	MHA			MSM	Rsv						MSP	MST
Power-ON default: 0000h															

Bit Number	Mnemonic	Description
Bits 31:27	Rsv	Reserved:
Bit 26	MRT	MstRxTx: Master Receiver/Transmitter This read/write bit specifies whether the Master has to receive data or to transmit data. Set this bit to 1 when the Master is to receive DATA. It does not matter if, as per the I ² C protocol, the Master has to transmit an ADDRESS to initiate a DATA read operation.
Bit 25	FNA	ForceNotAck: Force Not Acknowledged This read/write bit indicates the acknowledgement value to be driven on the SDA line in Master Receiver mode. In Master Transmitter mode it signifies an End of Data Transfer. 0: Master should acknowledge the received data 1: Master should NOT acknowledge the received data/End of Data Transfer for the Master Transmitter.
Bit 24	SAH	SlaveAck: Slave Acknowledge (active low) This read/write bit indicates if a Slave is to acknowledge the data transfer on the I ² C bus when functioning as a Receiver. 0: Slave should acknowledge received data 1: Slave should NOT acknowledge received data.
Bits 23:21	SAU	SleveAddUp: Upper Slave Address bits These three read/write bits contain the upper bits of the Slave Address. They are used if the Slave Addressing mode is 10-bit. The Slave acknowledges this address if detected on the I ² C bus.
Bits 20:14	SAL	SlaveAddLow: Lower Slave Address bits These read/write bits contain the lower seven bits of the Slave Address when using 7-bit address mode. The Slave acknowledges this address if detected on the I ² C bus.



I²C BUS CONTROLLER

Bit Number	Mnemonic	Description
Bit 13	SAS	SlvAddrSet: Slave Address Set This read/write bit should be set to indicate that the address to which the Slave is to respond has been programmed. A Slave cannot begin a transfer unless this bit is programmed to 1.
Bit 12	SAM	SlaveAddMode: Slave Address Mode This read/write bit indicates whether the I ² C Slave is addressed using the 7-bit or 10-bit extended address mode. 0: 7-bit addressing 1: 10-bit addressing
Bits 11:9	MHA	MstHsAddress: Master High-speed Address These read/write bits contain three bits of the address when the I ² C Controller is programmed to function as a High Speed Master.
Bit 8	MSM	MstSpeedMode: Master Speed Mode This read /write bit sets the Speed mode. 0: Master in Slow (100 kHz) or Fast (400 kHz) Mode 1: Master in High Speed Mode (3.4 MHz) When set to 1, and there is a START Command from the Host, the I ² C Controller drives a START on the I ² C Bus. The SCL frequency is changed according to the base divider value programmed in the CLOCK CONTROL register.
Bits 7:2	Rsv	Reserved
Bit 1	MSP	MstStop: Master STOP Read/write Master stop bit. When set to 1 it will stop the Master after the current operation has been completed. This is determined by the number of bytes to be transferred, as specified in the BYTE COUNT register. If the transfer is to be stopped immediately after the next byte, then the ForceNotAck bit should be set to 1.
Bit 0	MST	MstStart: Master START Read/write Master start bit. Set to 1 to cause the I ² C Master to drive a START condition on the I ² C bus lines if the ADDRESS byte to be transferred has been written to the Transmit FIFO. Otherwise, if the FIFO is empty, transmission starts soon after the address is written into the Transmit FIFO. This bit is self-clearing. Note: If the I ² C Controller is busy and the I ² C is acting as a Master, and in-between if START is given, the I ² C Master completes the current byte transmission or reception and immediately issues a START; the subsequent byte will then be treated as an address on the I ² C bus.

16.16.5. Control Register

Control

Offset 0x0Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv											MI	STNA	MANA	MTNA	MALI
Power-ON default: 0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBEI	BTIE	RBF	BRIE	RFS				TFS				Rsv	RFF	TFF	SR
Power-ON default: 7F80h															

Bit Number	Mnemonic	Description
Bits 31:21	Rsv	Reserved:
Bit 20	MI	MaskInterrupt: Setting this bit to 1 masks all interrupts even though they may be enabled.
Bit 19	STNA	SlvTmtNoAckIntEn: Slave Transmitter No Acknowledgement Interrupt Enabled If set to 1, then an interrupt is generated whenever a Slave Transmitter gets no acknowledgement.
Bit 18	MANA	MstAddrNoAckIntEn: Master Address No Acknowledgement Interrupt Enabled If set to 1, then an interrupt is generated whenever a Master transmitting an address gets no acknowledgement.
Bit 17	MTNA	MstTmtNoAckIntEn: Master Transmit No Acknowledgement Interrupt Enabled 0: No interrupt is generated 1: Generates an interrupt if the Master in transmission gets no acknowledgement.
Bit 16	MALI	MstArbLossIntEn: Master Arbitration Loss Interrupt Enabled If this bit is set, it enables interrupt generation if the Master loses arbitration.
Bit 15	TBEI	TmtBufferEmptyIntEn: Transmit Buffer Empty Interrupt Enabled If this bit is set, an interrupt is generated whenever the Transmitter FIFO is empty.

I2C BUS CONTROLLER

Bit Number	Mnemonic	Description
Bit 14	BEIE	ByteTmtIntEn: Byte Transmit Interrupt Enabled If this bit is set, an interrupt is generated whenever the byte in the Transmit FIFO is transmitted with the acknowledgement received.
Bit 13	RBF	RcvBufferFullIntEn: Receiver Buffer Full Interrupt Enabled If this bit is set, an interrupt is generated whenever the Receive Buffer is full.
Bit 12	BRIE	ByteRcvIntEn: Byte Receive Interrupt Enabled 0: Disable bit for interrupt generation for each byte received in the Receive FIFO 1: Enable bit for interrupt generation for each byte received in the Receive FIFO
Bits 11:8	RFS	RcvFifoSize: Receive FIFO Size These bits indicate the size of the Receive FIFO. 0: Size 1 1: Size 2 3: Size 3 and similarly for other FIFO sizes programmed.
Bits 7:4	TFS	TmtFifoSize: Transmit FIFO Size These bits indicate the size of the Transmit FIFO. 0: Size 1 1: Size 2 3: Size 3 and similarly for other FIFO sizes programmed.
Bit 3	Rsv	Reserved
Bit 2	RFF	RcvFifoFlush: (Receive FIFO Flush) Writing a 1 into this self clearing bit flushes the Receive FIFO.
Bit 1	TFF	TmtFifoFlush: (Transmit FIFO Flush) Writing a 1 into this self clearing bit flushes the Transmit FIFO.
Bit 0	SR	SoftReset: Writing a 1 into this self clearing bit resets the I ² C Controller.

16.16.6. Clock Control Register

Clock_Control

Offset 0x10h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSSU								SCLHi2							
Power-ON default: 9D00h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLHi								SCLLo							
Power-ON default: 9D9Dh															

Bit Number	Mnemonic	Description
Bits 31:24	SSSU	Start Stop Set Up: These read/write bits contain the value in multiples of system Clock of the Start and Stop delay.
Bits 23:16	SLCHi2	SCL High value 2: These read/write bits denote the SCL High value in multiples of system Clock Time Period to be driven on the I ² C Bus in HIGH SPEED Transfer Mode.
Bits 15:8	SCLHi	SCL High: These read/write bits denote the SCL High value in multiples of system Clock Time Period to be driven on the I ² C Bus, minimum value is 6.
Bits 7:0	SCLLo	SCL Low: These read/write bits denote the SCL Low value in multiples of system Clock Time Period to be driven on the I ² C Bus, minimum value is 6.

16.16.7. Scratch Register

Scratch

Offset 0x14h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Attributes															
Power-ON default: 0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Attributes															
Power-ON default: 0000h															

Bit Number	Mnemonic	Description
Bits 31:0		These read/write bits are used for setting ihold_ack and ihold_bit inputs of behavioural I ² C Slave, used to keep the SCL line pulled low during a clock acknowledgement and after each bit is transmitted on the SDA line.



16.16.8. Byte Count Register

Byte_Count

Offset 0x18h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Attributes															
Power-ON default: 0000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Attributes															
Power-ON default: 0000h															

Bit Number	Mnemonic	Description
Bits 31:0		These read/write bits are used to store the data transfer count. In Master Transmitter Mode, this signifies the number of bytes to be transferred by the I ² C Master. When in Master Receiver Mode, it signifies the number of bytes to be received.

17. POWER MANAGEMENT

17.1. INTRODUCTION

For full information on the action of the System Management Mode (SMM), please refer to the STMicroelectronics manual for the ST486 CPU Core. This chapter describes the SMM control registers for the STPC.

The STPC provides the following hardware structures to assist the software in managing system power consumption:

- System Activity detection,
- Three power-down timers,
- Doze timer for detecting short-duration lack of system activity,
- Standby timer for detecting medium-duration lack of system activity,
- Suspend timer for detecting long-term lack of system activity,
- House-keeping activity detection,
- House-keeping timer to cope with short bursts of house-keeping activity while dozing or in standby state,
- Peripheral Activity detection,
- Peripheral timer for detecting lack of peripheral activity,
- STPCLK# modulation to adjust system performance in various system power down states, including full power-on state.

Lack of system activity for progressively longer periods of times is detected by the three power-down timers. These timers can generate a System Management Interrupt (SMI) to the CPU so that the SMM software can put the system in decreasing states of power consumption. System activity in a power-down state can generate an SMI to allow the software to bring the system back up to the full power-on state. The chipset supports up to three power-down states: Doze state, Standby state and Suspend state. These correspond to increasing levels of power savings.

The chipset can detect the presence or absence of the following System activities:

- DMA Request (DRQ),
- Interrupt Request (INTR),
- Parallel IO (PIO),
- Serial IO (SIO) ,
- Keyboard (KBD),
- Floppy Disk Controller (FDC),
- Hard Disk Controller (HDC) ,
- PCI master device,
- Programmable address range.

Each of these can be individually enabled. The presence of an enabled system activity resets the power-down timers. The chipset generates the SMI when no system activity is detected for the delay period programmed in the power-down timers. The software can then put the appropriate sub-systems in the power-down mode, request STPCLK# assertion and stop CPU and other system clocks, program the current power-down state in the chipset and set up the next timer.

The presence of an enabled system activity, when the STPC is in a power-down state, will first enable any stopped clocks, wait for a programmable delay to allow any internal Phase Locked Loop (PLL) to stabilise and then deassert STPCLK# to enable CPU execution. The device can optionally generate an SMI to allow the SMM to bring the system back to the power-on state.

The current revision of the STPC does not implement support for stopping CPU and other system clocks.

In Doze or Standby state, a house-keeping activity, can bring the system back to full speed for a short period of time before returning back to Doze or Standby state. The chipset can detect the following house-keeping activities:

- DMA Request (DRQ),
- Interrupt Request (INTR),
- Keyboard (KBD),
- PCI master device.

The house-keeping timer determines the length of time the system will be on before returning to the original power-down state. An activity can be either a system activity or a house-keeping activity, but not both at the same time. Further, the Suspend state cannot make use of this feature.

The absence of the following peripheral activities can be enabled to cause an SMI and thus allow the software to put the unused peripherals in the power-down state, while the remainder of the system is still in full power-on state:

- Parallel IO (PIO),
- Serial IO (SIO) ,
- Keyboard (KBD),
- Floppy Disk Controller (FDC),
- Hard Disk Controller (HDC),
- A programmable address range.

Each of these can be individually enabled for inactivity detection. The presence of a peripheral activity does not reset the peripheral timer. It always times out after the programmed delay period. An SMI is generated if any enabled peripheral was not active for this time period. The device provides IO access trapping to detect access to a powered-down peripheral, so that the software can bring the peripheral to the power-on state before the access is completed.

The STPC can also carry out software transparent power management, if so enabled. In this mode of operation, doze and standby time-outs will change the CPU clock without generating an SMI. The state transitions from fully-on to doze or standby and back to fully-on will take place automatically. Also note that the suspend state can never be entered automatically but always requires software assist.

The STPC decodes the various activities listed below in [Table 17-1](#).

Table 17-1. Activity Detected

Activity	Detected via
ISA DMA masters	Low to high transition of hold request of 206
PCI masters	High to low transition of any of PCIIRQ2-0#
Parallel port	IO read/write at 378h-37Fh, 278h-27Fh and 3BCh-3BFh
Serial port	IO read/write at 3F8h-3FFh, 2F8h-2FFh, 3E8h-3EFh and 2E8h-2EFh
Keyboard	IO read/write at 60h, 62h, 64h and 66h
Floppy disk	IO read/write at 3F2h, 3F4h, 3F5h and 3F7h
Hard disk	IO read/write in 170h-177h, 376h,1F0h-1F7h and 3F6h address range as well as any bus master activity by the internal IDE controller.



17.2. POWER MANAGEMENT CONTROLLER REGISTERS

17.2.1. TIMER REGISTER 0

This register controls the timer for the selection of the length of timeout for the doze, standby and suspend modes.

Timer0				Access = 0022h/0023h				Regoffset = 060h			
7	6	5	4	3	2	1	0				
SUTT				STT				Rsv			
Default value after reset = undefined											

Bit Number	Mnemonic	Description
Bits 7-5	SUTT	<p>Suspend Timeout Timer, when set to any value other than the disable value (000), this timer will generate an SMI on time out.</p> <p>Once enabled, this timer counts down from the programmed value. If any of the enabled system activities are detected before time out, the timer will reset and start again. These bits are encoded as given in Table 17-2.</p> <p>The suspend timer will count whenever it is not disabled and the suspend time-out bit in the SMI status register 0 is not set to a 1.</p>
Bits 4-2	STT	<p>Standby Timeout Timer, when set to any value other than the disable value (000) this timer, on expiration, can either generate the SMI to the CPU or if programmed for auto-power saving (software transparent power management) mode, change the power-down state to Standby state (refer to auto-power saving mode for details of the power saving features that are enabled in the standby state). Similar to the Suspend timer, presence of an enabled system activity will reset the timer to restart counting. These bits are encoded as given in Table 17-3.</p> <p>The standby timer will count whenever it is not disabled and the standby time-out bit in the SMI status register 0 is not set to a 1.</p>
Bits 1-0	Rsv	Reserved.

Table 17-2. Suspend Timer Reset

Bit 7	Bit 6	Bit 5	Suspend Timer reset
0	0	0	Disabled
0	0	1	4 minutes
0	1	0	8 minutes
0	1	1	12 minutes
1	0	0	16 minutes
1	0	1	32 minutes
1	1	0	48 minutes
1	1	1	64 minutes



Table 17-3. Standby Timer Reset

Bit 4	Bit 3	Bit 2	Standby Timer reset
0	0	0	Disabled
0	0	1	Reserved
0	1	0	2 minutes
0	1	1	4 minutes
1	0	0	6 minutes
1	0	1	8 minutes
1	1	0	12 minutes
1	1	1	16 minutes



17.2.2. TIMER REGISTER 1

Timer1

Access = 0022h/0023h

Regoffset = 061h

7	6	5	4	3	2	1	0
Rsv	HKT			PTT			Rsv
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	Rsv	Reserved.
Bits 6-4	HKT	<p>House-keeping Timer. This timer determines how long the PMU will be in Doze house-keeping state when an enabled house-keeping activity is detected while in doze or standby power-down states. It is encoded as given in Table 17-4.</p> <p>The house-keeping counts only when the PMU is in one of the house-keeping states. Another house-keeping activity while the controller is in house_keeping state will reset the house-keeping timer to restart counting. A system activity detection in the house_keeping state will have the same effect as if the controller was in Doze or Standby state. Either an SMI will be generated to allow the software to bring the system to power-on state or the controller will automatically transition to power-on state. The house-keeping timer and function can be disabled by masking out all activity detection via the House-keeping Enable registers.</p>
Bits 3-1	PTT	<p>Peripheral Timeout Timer. When set to a value other than (000) this timer on expiration, will generate an SMI if any of the enabled peripherals remained inactive during the entire period. Unlike the power-down timers, the peripheral timer does not reset due to an enabled peripheral activity. It always times out after the programmed delay. An SMI is generated only if any of the enabled peripherals were inactive during this period. This field is encoded as given in Table 17-5.</p> <p>The peripheral timer counts whenever it is enabled.</p>
Bit 0	Rsv	Reserved.

Table 17-4. House-keeping Timer Reset

Bit 6	Bit 5	Bit 4	House-keeping Timer reset
0	0	0	Disabled
0	0	1	64 micro-seconds
0	1	0	128 micro-seconds
0	1	1	256 micro-seconds
1	0	0	Reserved
1	0	1	4 milli-seconds
1	1	0	16 milli-seconds
1	1	1	32 milli-seconds



Table 17-5. Peripheral Timer Reset

Bit 3	Bit 2	Bit 1	Peripheral Timer reset
0	0	0	Disabled
0	0	1	8 seconds
0	1	0	16 seconds
0	1	1	32 seconds
1	0	0	64 seconds
1	0	1	128 seconds
1	1	0	256 seconds
1	1	1	512 seconds



17.2.3. TIMER REGISTER 2

Timer 2

Access = 0022h/0023h

Regoffset = 08Dh

7	6	5	4	3	2	1	0
DTT			Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7- 5	DTT	Doze Timeout Timer. When set to any value other than the disable value (00), this timer, on expiration, can either generate the SMI to the CPU or if programmed for auto-power saving (software transparent power management) mode, change the power-down state to Doze state (refer to auto-power saving mode for details of the power saving features that are enabled with Doze state). Similar to the suspend timer, presence of an enabled system activity will reset the timer to restart counting. This 3-bit field is encoded as given in Table 17-6 . The doze timer will count whenever it is not disabled and the doze time-out bit in the SMI status register 0 is not set to a '1'.
Bits 4-2	Rsv	Reserved.

Table 17-6. Doze Timer Reset

Bit 7	Bit 6	Bit 5	Doze Timer reset
0	0	0	Disabled
0	0	1	50 milli-seconds
0	1	0	100 milli-seconds
0	1	1	500 milli-seconds
1	0	0	Reserved
1	0	1	4 seconds
1	1	0	8 seconds
1	1	1	16 seconds



17.2.4. SYSTEM ACTIVITY ENABLE REGISTER 0

This is the first of the three registers that control which system activity to detect.

Sys_Activ_en0			Access = 0022h/0023h			Regoffset = 062h	
7	6	5	4	3	2	1	0
DRQ	PCIM	PIO	SIO	KBD	FDC	HDC	Rsv
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	DRQ	DMA Request (DRQ).
Bit 6	PCIM	PCI master device (PCIM).
Bit 5	PIO	Parallel IO (PIO).
Bit 4	SIO	Serial IO (SIO).
Bit 3	KBD	Keyboard (KBD).
Bit 2	FDC	Floppy Disk Controller (FDC).
Bit 1	HDC	Hard Disk Controller (HDC).
Bit 0	Rsv	Reserved.

Programming notes:

When detected, the power-down timers will reload with their initial time values, or if enabled via the SMI control register, an SMI will be generated, or if programmed for auto-power down mode and in Doze or Standby power-down states, transition to power-on state will take place. Set the following bits to '1' to detect the associated activity, and to '0' to ignore the associated activity.



17.2.5. SYSTEM ACTIVITY ENABLE REGISTER 1

This is the second of the three registers that control which system activity to detect.

Sys_Activ_en1		Access = 0022h/0023h				Regoffset = 063h	
7	6	5	4	3	2	1	0
Rsv		AR0	Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	Rsv	Reserved. Must be programmed to '0'.
Bit 5	AR0	Address range 0.
Bits 4-0	Rsv	Reserved. Must be programmed to '0'.



17.2.6. SYSTEM ACTIVITY ENABLE REGISTER 2

This is the third of the three registers that control which system activity to detect.

Sys_Activ_en2			Access = 0022h/0023h			Regoffset = 064h	
7	6	5	4	3	2	1	0
IRQ15-1	IRQ0	NMI	Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	IRQ15-1	IRQ15-1 detection enabled.
Bit 6	IRQ0	IRQ0 detection enabled.
Bit 5	NMI	NMI detection enable.
Bits 4-0	Rsv	Reserved.



17.2.7. HOUSE-KEEPING ACTIVITY ENABLE REGISTER 0

This register controls which house-keeping activity to detect. House-keeping activities are detected only in Doze and Standby states. If enabled, a house-keeping activity reverts the system back to power-on state for a short period of time, programmed in the house-keeping timer. Set the following bits to a '1' to enable activity detection and a '0' to ignore the associated activity.

HK_Activ_en0			Access = 0022h/0023h			Regoffset = 065h	
7	6	5	4	3	2	1	0
DRQ	PCI MD	KBD	IRQ15-1	IRQ0	NMI	Rsv	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	DRQ	DMA Request (DRQ) activity
Bit 6	PCI MD	PCI master device activity
Bit 5	KBD	Keyboards (KBD) activity
Bit 4	IRQ15-1	IRQ15-1 activity
Bit 3	IRQ0	IRQ0 activity
Bit 2	NMI	NMI activity
Bits 1-0	Rsv	Reserved.



17.2.8. HOUSE-KEEPING ACTIVITY ENABLE REGISTER 1

This is the second house-keeping activity detection enable register.

HK_Activ_en1		Access = 0022h/0023h				Regoffset = 066h	
7	6	5	4	3	2	1	0
Rsv		AR0	Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	Rsv	Reserved. Must be programmed to '0'.
Bit 5	AR0	Address range 0.
Bits 4-0	Rsv	Reserved. Must be programmed to '0'.



17.2.9. PERIPHERAL INACTIVITY DETECTION REGISTER 0

This register controls which peripheral inactivity is enabled for generating an SMI on a peripheral time-out.

Perif_Inact0			Access = 0022h/0023h			Regoffset = 067h	
7	6	5	4	3	2	1	0
PIO	SIO	KBD	FDC	HDC	ARO	Rsv	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	PIO	Parallel IO (PIO) activity.
Bit 6	SIO	Serial IO (SIO) activity.
Bit 5	KBD	Keyboard (KBD) activity.
Bit 4	FDC	Floppy Disk Controller (FDC) activity.
Bit 3	HDC	Hard Disk Controller (HDC) activity.
Bit 2	ARO	Address range 0.
Bits 1-0	Rsv	Reserved. Must be programmed to '0'.

Programming notes:

Lack of peripheral activity for an enabled peripheral for one peripheral time-out period generates an SMI. A '1' in a bit position enables the SMI generation for the associated peripheral and a '0' disables it. Software can use the Peripheral Inactivity status register to determine which peripheral should be powered down.



17.2.10. PERIPHERAL ACTIVITY DETECTION REGISTER 0

This register controls which peripheral accesses will cause an SMI.

Perif_Act0			Access = 0022h/0023h			Regoffset = 069h	
7	6	5	4	3	2	1	0
PIO	SIO	KBD	FDC	HDC	ARO	Rsv	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	PIO	Parallel port (PIO) access
Bit 6	SIO	Serial port (SIO) access
Bit 5	KBD	Keyboard (KBD) access
Bit 4	FDC	Floppy Disk Controller (FDC) access
Bit 3	HDC	Hard Disk Controller (HDC) access
Bit 2	ARO	Address range 0
Bits 1-0	Rsv	Reserved. Must be programmed to '0'

Programming notes:

Typically the power management software will detect non-usage of a peripheral device via Peripheral in-activity status registers, bring the peripheral into power down state and then enable trapping access to that peripheral via this register.

Thus when an application attempts to make use of a powered down peripheral, the access is trapped and an SMI is generated to allow software to re-power the peripheral device before allowing the access to complete. This register is the first of the two such registers.

A '1' in a bit position enables SMI generation for the associate peripheral and a '0' disables.



17.2.11. PERIPHERAL ACTIVITY DETECTION REGISTER 1

This is the second register that controls which peripheral accesses will cause an SMI. This register is similar in functionality to Peripheral Activity detection register 0.

Perif_Act1			Access = 0022h/0023h			Regoffset = 06Ah	
7	6	5	4	3	2	1	0
Rsv							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	Rsv	Reserved. Must be programmed to '0'.



17.2.12. ADDRESS RANGE 0 REGISTER 0

This register contains bits which are compared with PCI address bits 31-24 if range compare is enabled for memory cycle or compared against bits 15-8 if range compare is enabled for IO cycles.

Add_Rang0-0			Access = 0022h/0023h			Regoffset = 06Bh	
7	6	5	4	3	2	1	0
Default value after reset = 00h							



17.2.13. ADDRESS RANGE 0 REGISTER 1

Add_Rang0-1				Access = 0022h/0023h				Regoffset = 06Ch			
7	6	5	4	3	2	1	0				
Default value after reset = 00h											

Bit Number	Mnemonic	Description
Bits 7-3		These bits are compared with PCI address bits 23-19 if range compare is enabled for memory cycle, or compared against bits 7-3 if range compare is enabled for IO cycles.
Bit 2		This bit is compared with PCI address bit 18 if range compare is enabled for memory cycle, or compared with address bit 2 if range compare is enabled for IO cycles and range is 4-Bytes. Otherwise this bit when 1 specifies that the range of IO address to be compared is 16-Bytes and when 0, the range is 8-Bytes.
Bit 1		This bit is compared with PCI address bit 17 if range compare is enabled for memory cycle. Otherwise if range compare is enabled for IO cycles, this bit if 1 specifies that the range of IO address to be compared is 8/16-Bytes and when 0 the range is 4-Bytes.
Bit 0		This bit when '1' specifies that range compare should be done for memory cycles and when '0', for IO cycles.



17.2.14. SMI CONTROL REGISTER 0

This register controls the generation of an SMI, as follows:

SMI_Cont0				Access = 0022h/0023h			Regoffset = 071h	
7	6	5	4	3	2	1	0	
							Rsv	
Default value after reset = 00h								

Bit Number	Mnemonic	Description
Bit 7		If '1' then generate SMI on Doze time-out. Otherwise if set to a '0', the hardware will transition to Doze state automatically on Doze time-out.
Bit 6		If '1' then generate SMI on Standby time-out. Otherwise if set to a '0', the hardware will transition to Standby state automatically on Standby time-out.
Bit 5		If '1' then generate SMI on Suspend time-out. Otherwise if set to a '0', SMI is not generated. The hardware never transitions into Suspend state by itself.
Bit 4		If '1' then generate SMI on House-keeping time-out. Otherwise if set to a '0', the hardware will automatically transition back to the doze or standby state (which ever state it was in before entering house-keeping state).
Bit 3		If '1' then generate SMI on detecting a house-keeping activity. Otherwise if set to a '0', and if in Doze or Standby state, the hardware will automatically transition to the associated house_keeping states for the duration programmed in the house-keeping timer.
Bit 2		If '1' then generate SMI on detecting a system activity. Otherwise if set to a '0', and if in Doze or Standby state, the hardware will automatically transition to Power-on state on detecting a unmasked system activity. This bit will typically be set to a '1' by software on entering a power-down state so that a system activity can wake up the system.
Bit 1		This is a write only bit. Setting this bit to a '1' sets bit-7 of the SMI status register 1 and generates an SMI. This bit however will always read back as '0'.
Bit 0	Rsv	Reserved.

17.2.15. SMI STATUS REGISTER 0

This register contains the status information pertaining to the SMI.

SMI_Stat0

Access = 0022h/0023h

Regoffset = 073h

7	6	5	4	3	2	1	0
DTO	STO	STO	HKT	HKA	SAD	PID	PAD
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	DTO	Doze time-out. This bit is set to a '1' when Doze time-out occurs. An SMI will be generated if associated SMI enable bit in SMI Control register is set to a '1'. The software must write this bit to a '1' to deassert SMI#. If SMI generation has been disabled then the controller will automatically transition to Doze state. This bit will then be cleared on transition from Doze or Standby to Power-on state.
Bit 6	STO	Standby time-out. This bit will be set to a '1' when Standby time-out occurs. An SMI will be generated if the associated SMI enable bit in SMI Control register is set to a '1'. The software must write this bit to a '1' to deassert SMI#. If SMI generation has been disabled then the hardware will automatically transition to Standby state. This bit will then be cleared on transition Standby to Power-on state.
Bit 5	STO	Suspend time-out. This bit will be set to a '1' when Suspend time-out occurs. An SMI will be generated if the associated SMI enable bit in the SMI Control register is set to a '1'. The software must write this bit to a '1' to deassert SMI#.
Bit 4	HKT	House-keeping timeout detected. This bit will be set to a '1' if the controller is in one of the house-keeping states and the house-keeping timer expires. An SMI will be generated if the associated SMI enable bit in the SMI Control register is set to a '1'. The software must write this bit to a '1' to deassert SMI#. If the SMI generation has been disabled, the hardware will automatically transition to doze or standby state. This bit then will be cleared on transition from Doze or Standby states to any other state.
Bit 3	HKA	House-keeping activity detected. This is a read-only bit and represents the OR of the System activity status registers masked (ANDed) with the corresponding bits in the House-keeping Activity enable registers. An SMI will be generated when this bit is a '1' and if the associated SMI enable bit in the SMI Control register is set to a '1'. The software can refer to Activity status register to determine the cause of the interrupt. The software must clear the corresponding bits of the Activity Status register to deassert SMI#. If SMI generation has been disabled and if the controller in Doze or Standby state, it will automatically transition to House-keeping state.

Bit Number	Mnemonic	Description
Bit 2	SAD	System Activity detected. This is a read-only bit and represents the OR of the System activity Status registers masked (ANDed) with the corresponding bits of the System Activity enable registers. An SMI will be generated if this bit is a '1' and if the associated SMI enable bit in SMI Control register is set to a '1'. The software can refer to Activity status register to determine the cause of this interrupt. The software must clear the System Activity Status registers bits for the enabled system activities to deassert SMI#. If SMI generation has been disabled and if the controller is in Doze or Standby state, it will automatically transition to Power-on state.
Bit 1	PID	Peripheral Inactivity detected. This is a read-only bit and represents the OR of Peripheral Inactivity Status register bits masked (ANDed) with the associated Peripheral inactivity detection register bit. An SMI# will be generated when this bit is a '1'. The software can refer to Peripheral Inactivity status registers to determine which peripheral should be powered down. The software must clear the corresponding bits of the Peripheral Inactivity detection register to deassert SMI#.
Bit 0	PAD	Peripheral Activity Detected. This is a read-only bit and represents the OR of the System activity Status register masked (ANDed) with the corresponding bits of the Peripheral Activity detection registers. An SMI will be generated when this bit is a '1'. The software can refer to the System Activity status register to determine which peripheral caused the interrupt. The software must clear the corresponding bits of the System activity register to deassert SMI#.

Programming notes:

The SMI# output is a logical OR of all the bits (ANDed with their respective SMI generation enable bits) in this register. SMI# output will be deasserted within 3 PCI clocks after the cause of the SMI# is cleared. This register defaults to 00h after reset deasserting SMI# output.



17.2.16. SMI STATUS REGISTER 1

This register is similar to SMI Status register 0 in that it reports the cause of the SMI to the software.

SMI_Stat1				Access = 0022h/0023h			Regoffset = 074h	
7	6	5	4	3	2	1	0	
S SMI	Rsv							
Default value after reset = 00h								

Bit Number	Mnemonic	Description
Bit 7	S SMI	Software SMI. This bit is set to a '1' by write writing a '1' in bit-1 of the SMI Control register. The software must clear this bit to deassert SMI#.
Bits 6-0	Rsv	Reserved.



17.2.17. PERIPHERAL INACTIVITY STATUS REGISTER 0

This register contains a ‘1’ in a bit position if the associated peripheral was inactive for the entire duration of the last peripheral time-out period.

Perif_Stat0			Access = 0022h/0023h			Regoffset = 075h	
7	6	5	4	3	2	1	0
PIO	SIO	KBD	FDC	HDC	ARO	Rsv	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	PIO	Parallel port (PIO) activity.
Bit 6	SIO	Serial port (SIO) activity.
Bit 5	KBD	Keyboard (KBD) activity.
Bit 4	FDC	Floppy Disk Controller (FDC) activity.
Bit 3	HDC	Hard Disk Controller (HDC) activity.
Bit 2	AR0	Address range 0
Bits 1-0	Rsv	Reserved.

It can also be cleared by software by writing a ‘1’ in the bit which is set to ‘1’.

Programming notes:

A bit in this register is set to a ‘1’ only at peripheral timer time-out. It is set to a ‘0’ as soon as an activity from the associated peripheral is detected.

The status reflected in this register is not conditioned by whether or not the peripheral was enabled for in-activity detection through the Peripheral Inactivity Detection registers. The SMI however will be generated only if any of the enabled peripherals (via Peripheral Inactivity Enable register) were inactive for the entire duration of the peripheral time out.



17.2.18. ACTIVITY STATUS REGISTER 0

This register records presence of activity.

Activ_Stat0			Access = 0022h/0023h			Regoffset = 077h	
7	6	5	4	3	2	1	0
DRQ	PCIM	PIO	SIO	KBD	FDC	HDC	Rsv
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	DRQ	DMA Request (DRQ) activity.
Bit 6	PCIM	PCI master device (PCIM) activity.
Bit 5	PIO	Parallel IO (PIO) activity.
Bit 4	SIO	Serial IO (SIO) activity.
Bit 3	KBD	Keyboard (KBD) activity.
Bit 2	FDC	Floppy Disk Controller (FDC) activity.
Bit 1	HDC	Hard Disk Controller (HDC) activity.
Bit 0	Rsv	Reserved.

Programming notes:

A '1' in a bit position indicates that presence of the associated activity since the bit was last cleared. Once set, a bit of this register can only be cleared by software writing a '1' to it or by reset or if auto power management is enabled then any transition to Doze or Standby state (including the ones from house-keeping states) will clear all enabled System and House-keeping activities.

The status reflected in this register is not conditioned by the settings of System Activity Enable, House-keeping Activity Enable, Peripheral Inactivity or Peripheral Activity Detection registers.



17.2.19. ACTIVITY STATUS REGISTER 1

This register is similar to Activity Status register 0. It contains the status for the following bits.

Activ_Stat1		Access = 0022h/0023h				Regoffset = 078h	
7	6	5	4	3	2	1	0
Rsv		AR0	Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	Rsv	Reserved. Must be programmed to '0'
Bit 5	AR0	Address range 0
Bits 4-0	Rsv	Reserved. Must be programmed to '0'



17.2.20. ACTIVITY STATUS REGISTER 2

This register is similar to Activity Status registers 0 and 1.

Activ_Stat2			Access = 0022h/0023h			Regoffset = 079h	
7	6	5	4	3	2	1	0
IRQ15-1	IRQ0	NMI	Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	IRQ15-1	IRQ15-1 activity.
Bit 6	IRQ0	IRQ0 activity.
Bit 5	NMI	NMI activity.
Bits 4-0	Rsv	Reserved.



17.2.21. PMU STATUS REGISTER

This register contains the state the power management controller currently is in.

PMU		Access = 0022h/0023h				Regoffset = 07Ah	
7	6	5	4	3	2	1	0
Rsv	PMU	PMU	PMU	PMU	PMU	PMU	PMU
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	Rsv	Reserved.
Bit 6	PMU	PMU microsecond clock test mode. This bit is for factory use only and must be set to a '0' by the software. Setting this bit to a 1' causes the microsecond clock to tick at oscillator clock frequency instead of every microsecond.
Bit 5	PMU	PMU millisecond clock test mode. This bit is for factory use only and must be set to a '0' by the software. Setting this bit to a 1' causes the millisecond clock to tick at oscillator clock frequency instead of every millisecond.
Bit 4	PMU	PMU second clock test mode. This bit is for factory use only and must be set to a '0' by the software. Setting this bit to a 1' causes the second clock to tick at oscillator clock frequency instead of every second.
Bit 3	PMU	PMU minute clock test mode. This bit is for factory use only and must be set to a '0' by the software. Setting this bit to a 1' causes the minute clock to tick at oscillator clock frequency instead of every minute.
Bit 2	PMU	PMU state (see Table 17-7).
Bit 1	PMU	PMU state (see Table 17-7).
Bit 0	PMU	PMU state (see Table 17-7).

Table 17-7. PMU State

Bit 2	Bit 1	Bit 0	PMU state
0	0	0	Power-on
0	0	1	Doze
0	1	0	Standby
0	1	1	Suspend
1	0	1	Doze_house_keeping
1	1	0	Standby_house_keeping
1	1	1	Reserved



The architecture allows for: (1) the software to explicitly program the power-down state of the controller, or (2) the controller can change states automatically (auto-power down mode of operation), or (3) a mix of the two. Some power-down states are entered and exited automatically by the hardware, while others require software assist. This is based on the SMI Control register settings as follows:

Transition from Power-on to Doze state will take place automatically on Doze time-out, if bit-7 of the SMI control register is set to a '0'. Otherwise if bit-7 is programmed to be a '1', an SMI will be generated and the software can change the state to Doze.

Transition from Doze to Power-on will take place automatically in the presence of an enabled system activity if bit-2 of the SMI control register is programmed to '0'. Otherwise if bit-2 is programmed to a '1', an SMI will be generated and software can change the state to Power-on.

Transition from Doze to Doze_house_keeping state will take place automatically if an enabled house_keeping activity is detected and bit-4 of the SMI control register is set to a '0'. Otherwise if bit-7 is programmed to be a '1', an SMI will be generated.

Transition from Doze_house_keeping state to Doze will take place automatically on house-keeping time-out if bit-3 of the SMI control register is set to a '0'. Otherwise an SMI will be generated.

Transition from Doze_house_keeping state to Power-on will take place on detecting an enabled system activity automatically if bit-2 of the SMI control register is programmed to '0'. Otherwise an SMI will be generated.

Transitions from Doze or Power-on state to Standby will take place automatically on standby time-out if bit-6 of the SMI control register is set to a '0'. Otherwise an SMI will be generated.

Transition from Standby to Power-on will take place automatically in presence of an enabled system activity if bit-2 of the SMI control register is programmed to be a '0'. Otherwise if bit-2 is programmed to a '1', an SMI will be generated and software can change the state to Power-on.

Transition from Standby to Standby_house_keeping state will take place automatically if an enabled house_keeping activity is detected and bit-4 of the SMI control register is set to a '0'. Otherwise if bit-7 is programmed to be a '1', an SMI will be generated.

Transition from Standby_house_keeping state to Standby will take place automatically on house-keeping time-out if bit-3 of the SMI control register is set to a '0'. Otherwise an SMI will be generated.

Transition from Standby_house_keeping state to Power-on will take place on detecting an enabled system activity automatically if bit-2 of the SMI control register is programmed to '0'. Otherwise SMI interrupt will be generated.

The hardware never transitions to Suspend state automatically.

The power saving features associated with each power-down state are independent of how the state was entered.

17.2.22. GENERAL PURPOSE REGISTER

This is a read/write IO register that can be used by software.

GP		Access = 0022h/0023h				Regoffset = 07Bh	
7	6	5	4	3	2	1	0
GP	GP	GP	GP	GP	GP	GP	GP
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	GP	General Purpose Register Bit 7.
Bit 6	GP	General Purpose Register Bit 6.
Bit 5	GP	General Purpose Register Bit 5.
Bit 4	GP	General Purpose Register Bit 4.
Bit 3	GP	General Purpose Register Bit 3.
Bit 2	GP	General Purpose Register Bit 2.
Bit 1	GP	General Purpose Register Bit 1.
Bit 0	GP	General Purpose Register Bit 0.

Programming notes:

Writing to this register also updates the external '373 latch that can be used to control external devices for power-down purposes. Reads of this register return the value of this internal register.

The GPIOCS# signal will be asserted when writing to this register to latch the data on the ISA data bus.



17.2.23. CLOCK CONTROL REGISTER 0

This register allows control over power saving via stop clock modulation. The power-saving can be tuned to the power-management state the PMU is in.

Clk_Cont0				Access = 0022h/0023h			Regoffset = 07Ch	
7	6	5	4	3	2	1	0	
STPCLK			DSSS			STPCLK	Rsv	
Default value after reset = 00h								

Bit Number	Mnemonic	Description
Bits 7-5	STPCLK	Power-on and housekeeping states STPCLK# modulation control. These bits control the duty cycle of STPCLK# deassertion when the PMU is in Power-on or one of the house-keeping states given in Table 17-8 . The STPCLK# is deasserted and the duty-cycle control ignored if an SMI is pending.
Bit 4-2	DSSS	Doze/Standby/Suspend states STPCLK# modulation control. These bits control the duty cycle of the STPCLK# deassertion when PMU is in one of the power-down states as given in Table 17-9 . The STPCLK# is deasserted and the duty-cycle control ignored if an SMI is pending.
Bit 1	STPCLK	STPCLK# modulation period. If '1' then the period is 64ms else, if '0', then the period is 64μs.
Bit 0	Rsv	Reserved.

Table 17-8. Power-on and Housekeeping States

Bit 7	Bit 6	Bit 5	Ratio	Power-on STPCLK# Modulation
0	0	0	1	STPCLK# is never asserted
0	0	1	1/2	1 half period
0	1	0	1/4	1 quarter period
0	1	1	1/8	one-eighth period
1	0	0	1/16	one-sixteenth period
1	0	1	1/32	1/32 period
1	1	0	1/64	1/64 period
1	1	1		Reserved.



Table 17-9. Doze/Standby/Suspend States

Bit 4	Bit 3	Bit 2	Ratio	Doze STPCLK# Modulation
0	0	0	1	STPCLK is never asserted
0	0	1	1/2	1 half period
0	1	0	1/4	1 quarter period
0	1	1	1/8	one-eighth period
1	0	0	1/16	one-sixteenth period
1	0	1	1/32	1/32 period
1	1	0	1/64	1/64 period
1	1	1	0	The entire period



17.2.24. DOZE TIMER READ BACK REGISTER

This read only register is provided for test purposes to read back the current value of the upper 8-bits of the 9-bit doze timer.

Doze				Access = 0022h/0023h				Regoffset = 088h			
7	6	5	4	3	2	1	0				
Default value after reset = 00h											

Bit Number	Mnemonic	Description
Bits 7-0		Bits 8-1 of the current value of the doze timer.

Programming notes:

This register should not be used by the software.

Note that bit 0 of the current value of the doze timer is not readable.



17.2.25. STANDBY TIMER READ BACK REGISTER

This read only register is provided for test purposes to read back the current value of 5-bit standby timer.

Standby			Access = 0022h/0023h			Regoffset = 089h	
7	6	5	4	3	2	1	0
Rsv							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-5	Rsv	Reserved.
Bits 4-0		Bits 4-0 of the current value of the standby timer.

Programming notes:

This register should not be used by software.



17.2.26. SUSPEND TIMER READ BACK REGISTER

This read only register is provided for test purposes to read back the current value of the 7-bit Suspend timer.

Suspend			Access = 0022h/0023h			Regoffset = 08Ah	
7	6	5	4	3	2	1	0
Rsv							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	Rsv	Reserved.
Bits 6-0		Bits 6-0 of the current value of the suspend timer.

Programming notes:

This register should not be used by software.



17.2.27. HOUSE-KEEPING TIMER READ BACK REGISTER

This read only register is provided for test purposes to read back the current value of the upper 8-bits of the 9-bit house-keeping timer.

HK_Timer				Access = 0022h/0023h				Regoffset = 08Bh			
7	6	5	4	3	2	1	0				
Default value after reset = undefined											

Bit Number	Mnemonic	Description
Bits 7-0		Bits 8-1 of the house-keeping timer.

Programming notes:

This register should not be used by software.



17.2.28. PERIPHERAL TIMER READ BACK REGISTER

This read only register is provided for test purposes to read back the current value of the upper 8-bits of the 9-bit Peripheral timer.

Perif_Timer			Access = 0022h/0023h			Regoffset = 08Ch		
7	6	5	4	3	2	1	0	
Default value after reset = undefined								

Bit Number	Mnemonic	Description
Bits 7-0		Bits 8-1 of the Peripheral timer.

Programming notes:

This register should not be used by software.



Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

© 2000 STMicroelectronics - All Rights Reserved

The ST logo is a registered trademark of STMicroelectronics.

All other names are the property of their respective owners.

STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - France - Germany - Italy - Japan - Korea - Malaysia - Malta - Mexico - Morocco - The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.

